Globale Optimierung extrem aufwendiger Funktionen mit hochparallelen und sequentiellen Methoden

Vom Fachbereich Mathematik

der Technischen Universität Darmstadt

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

genehmigte

Dissertation

von

Dipl. Math. Michael Godzierz

aus Bensheim

Referent:	Prof. Dr. M. Kiehl
Koreferentin:	Prof. Dr. M. Dür
Tag der Einreichung:	11. Juni 2007
Tag der mündlichen Prüfung:	16. Juli 2007

Darmstadt 2007
D 17

Inhaltsverzeichnis

Einleitung

Praktische Problemstellung

Die ursprüngliche Motivation dieser Arbeit ist gegeben durch ein Konzept aus dem Katalysatorbau. Wir betrachten einen Rohrreaktor wie in Abbildung 1. In das eine Ende wird ein Reaktionsgemisch, z.B. Abgase, hineingeblasen, es findet eine Reaktion statt, die durch an der Rohrinnenwand aufgebrachte Stoffe katalysiert wird. Eventuell muß man das Rohr erwärmen, um die Reaktion anzustoßen. Das zweite Ende des Rohres ist oft so plaziert, daß zwischen Ein- und Ausgangsbereich ein Wärmeaustausch stattfindet. Damit wird das eintretende Gas erhitzt und das austretende abgekühlt.

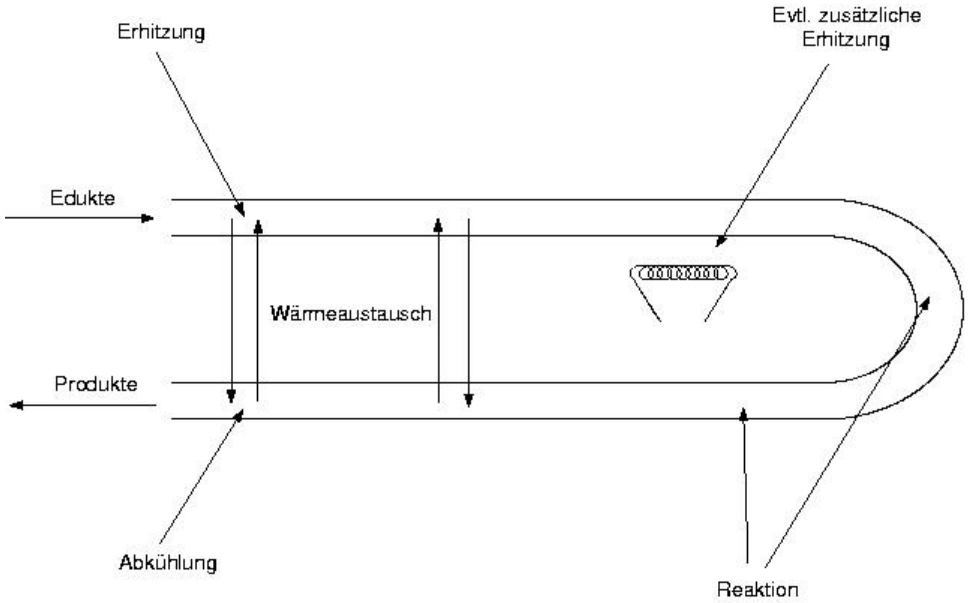


Abbildung 1: Rohrreaktor

Bei einem Feststoffkatalysator ist die katalytische Wirkung von der Oberfläche abhängig. Daher ersetzt man z.B. ein großes Rohr durch ein Bündel aus vielen (wir gehen von etwa 1000 aus) kleinen Rohren mit sechseckigem Querschnitt. Siehe dazu Abbildung 2.

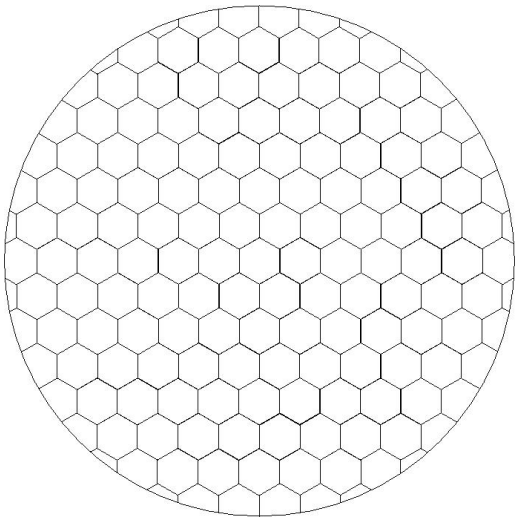


Abbildung 2: Katalysatorquerschnitt

Es soll nun die Effizienz eines solchen Katalysators optimiert werden. Diese hängt von produktionstechnischen Parametern ab. Also etwa von der genauen Zusammensetzung der Legierung der Rohrinnenseite, wir gehen dabei von maximal 10 Bestandteilen aus. Aber auch davon, auf welchem Wege legiert wurde. So ist die Korngröße einer Metallegierung abhängig von der Schmelztemperatur, aber auch von der Geschwindigkeit der Erhitzung bzw. Abkühlung. Diese haben daher Einfluß auf die Effizienz des Endproduktes.
Der Effizienztest eines solchen Katalysators ist extrem teuer. Zum einen muß der teure Prototyp erst gebaut werde, weiterhin muß er lange getestet werden, damit man ihn auch zuverlässig auf seinen Verschleiß hin prüfen kann. Ohne bedeutenden Mehraufwand kann jedoch ein Reaktor gebaut und getestet werden, der aus mit verschiedenen Legierungen beschichteten Röhrchen zusammengesetzt ist. Dazu muß etwa vor dem Schmelzprozeß in jede Rohrleitung ein anderes Metallpulvergemisch eingebracht werden. Es kann jedoch nicht ohne Mehraufwand ein Reaktor gebaut werden, der aus Röhrchen zusammengesetzt ist, die unterschiedlich erhitzt bzw. abgekühlt wurden. Dies berücksichtigen wir im mathematischen Teil aber nicht. Im Betrieb wird dann jedes einzelne Röhrchen für sich, z.B. mit einer dafür geeigneten Sonde, getestet. Es kann also parallel mit nahezu gleichem

Aufwand die Effektivität von etwa 1000 verschiedenen Katalysatoren getestet werden.

Mathematischer Sachverhalt

Mathematisch gesehen haben wir folgendes Problem zu lösen: Gegeben sei $f : \mathbb{R}^n \to \mathbb{R}$ und $\mathcal{D} = [0, 1]^n$. f sei auf $\mathcal{D}$ stetig.
Es gilt nun, das Problem

$$\min_{x \in \mathcal{D}} f(x)$$

zu lösen, wobei nicht irgendein lokales, sondern wirklich ein globales Minimum gemeint ist. Dies ist aber noch nicht ungewöhnlich.
Unsere Zielfunktion zeichnet sich jedoch dadurch aus, daß sie nicht als analytischer Ausdruck gegeben ist. Auswertungen können nur sehr mühsam beschafft werden, Ableitungen gar nicht. Dies ist sogar derart aufwendig, daß die Rechenzeit numerischer Verfahren, die neue Auswertungspunkte generieren sollen, vernachlässigbar ist. Die Verfahren, die wir suchen, müssen also unbedingt die Eigenschaft haben, mit möglichst wenigen Funktionsauswertungen auszukommen. Dabei spielen sonstige Rechnungen keine Rolle.
Wir dürfen weiterhin ausnutzen, daß die parallele Auswertung an bis zu 1000 Stellen nur unwesentlich teurer ist als die Auswertung in einer Stelle. Aus der Problemstellung geht hervor, daß maximal 10 Eingangsdaten derart hochparallel verändert werden können, also gehen wir davon aus, daß $n \leq 10$ gilt.
Leider ist es nicht möglich, Ableitungsinformationen der Zielfunktion aus dem experimetellen Aufbau direkt zu gewinnen. Dies schließt jedoch nicht aus, daß Ableitungen existieren und wir sie auf numerischem Wege beschaffen.
Die eben beschriebene Fragestellung ist auch ohne den Zusatz des hochparallelen Auswertens interessant. In der Literatur wurde darauf schon eingegangen. Diese Arbeit beschäftigt sich mit beiden Problemstellungen. Hat die Zielfunktion die Eigenschaft ohne Mehraufwand parallel an viele Stellen ausgewertet werden zu können, so sprechen wir in dieser Arbeit von der parallelen Problemstellung. Ist dies nicht möglich, kostet also jede einzelne Auswertung extrem viel und ist damit die parallele Auswertung entsprechend teurer, dann sprechen wir von der sequentiellen Problemstellung.
Die in der Literatur erwähnten Verfahren, die zur Bearbeitung der sequentiellen Problemstellungen entwickelt wurden, nutzen nicht aus, daß wir ohne nennenswerten Mehraufwand hochparallel auswerten können. Auch gibt es keine offen-

sichtliche Parallelisierung ohne erheblichen Mehraufwand bzw. ohne grundlegende Veränderung der jeweiligen Methode. Deswegen kann man die parallele Problemstellung als neuartig ansehen.
In dieser Arbeit soll ein Verfahren für die parallele Problemstellung entwickelt werden. Nach Möglichkeit soll es auch für die sequentielle Problemstellung einsetzbar sein. Die Art der Randbedingungen war bei der Entwicklung zweitrangig, es ging vielmehr darum, im Inneren eines einfach strukturierten Suchgebietes möglichst effiziente Methoden zu finden. Deshalb beschränken wir uns auf $\mathcal{D} = [0,1]^n$. Die Verallgemeinerung auf andere Gebiete ist rein technisch.

Leitgedanken der entwickelten Methoden

Der Aufbau dieser Arbeit ist verständlicher, wenn die in Kapitel 4 neu entwickelten Verfahren kurz erläutert werden. Dadurch erkennt man, welche Techniken wesentlich sind und warum diese in den Kapiteln 1, 2 und 3 vorgestellt werden. Da man das Lösungsverfahren für die sequentielle Problemstellung als Sonderfall der Methode für die parallele Problemstellung auffassen kann, erklären wir die wesentlichen Ansätze der zweitgenannten Prozedur. Wir gehen dabei aber nicht auf Details ein.
Grob gesagt steht man in jedem Verfahrensschritt vor der Aufgabe, in jedem Schritt etwa 1000 Auswertungspunkte sinnvoll auf die zulässige Menge $\mathcal{D}$ zu verteilen. Dabei soll beachtet werden, daß einerseits im Verfahrensverlauf keine großen Lücken in $\mathcal{D}$ entstehen, andererseits dort, wo sich gute Zielfunktionswerte befinden, genauer gesucht wird um möglichst schnell große Fortschritte zu erzielen. Man hat also ein globales Abtasten mit lokaler Suche zu kombinieren.
Das globale Absuchen erfolgt in den neuen Verfahren nach folgendem Prinzip: In Abhängigkeit von den bisher vorhandenen Auswertungspunkten und den dort vorliegenden Zielfunktionswerten wird untersucht, wie plausibel es ist, daß in einer bestimmten zusätzlichen Stelle $x^z \in \mathcal{D}$ der Zielfunktionswert y^z erreicht wird. y^z wird so gewählt, daß damit auf jeden Fall eine Verbesserung im Vergleich zum bisher besten Zielfunktionswert erreicht würde. Die Idee ist, einen Interpolanden Φ^z, der die bisherigen Stützstellen und (x^z, y^z) gleichzeitig interpoliert, zu errechnen und ihn daraufhin zu prüfen, wie sehr er „gewunden“ ist. Kommt man ohne große Krümmungen aus, sieht man es als wahrscheinlich an, daß in der Stelle x^z der Wert y^z erreicht werden kann. Ist Φ^z dagegen stark gekrümmt, ist die Stelle x^z uninteressant.
Wir sind somit auf Resultate aus der Interpolationstheorie angewiesen. Die bekanntesten Ansätze werden in Kapitel 1 vorgestellt und im Hinblick auf ihre

Brauchbarkeit für unsere Zwecke untersucht. Mit diesem Wissen versteht man auch die bisher bekannten Verfahren, die für die sequentielle Problemstellung entwickelt wurden. Sie werden in Kapitel 2 zusammengefaßt.
Weiterhin ist in die Methode zur Bearbeitung der parallelen Problemstellung ein lokales Suchverfahren integriert. Dieses beinhaltet zum einen die Minimierung des Interpolanden zu den bisherigen Auswertungen. Also benötigt man erneut die Interpolationstheorie aus Kapitel 1. Zum anderen erlaubt die parallele Problemstellung die Durchführung einer Verallgemeinerung des Newton-Verfahrens. Diese wird in Kapitel 3 beschrieben.

Aufbau der Arbeit

Die Arbeit ist in fünf Kapitel gegliedert: Im ersten werden die für uns wichtigsten Resultate aus der Interpolationstheorie zusammengestellt und erklärt, warum vieles Naheliegende für uns nicht in Frage kommt:
Interpolation mit Polynomen wäre für uns vom Ansatz her sehr kompliziert, langsam auf dem Rechner und letztendlich zu unflexibel.
Abschnittsweise Polynome sind auch relativ aufwendig, insbesondere wenn man mehr als stetige Differenzierbarkeit fordert. Es sind nur sinnvolle Ansätze für 2-D bekannt. Alles, was darüber hinausgeht, ist sehr kompliziert und offensichtlich ungeeignet.
Shepard Methoden zeichnen sich zwar im Beliebigdimensionalen durch ihre Einfachheit und Flexibilität aus, sind aber entweder nicht differenzierbar oder haben immer Gradientennullstellen in den Interpolationsknoten. Um dies zu beseitigen, bräuchte man Gradienteninformation über die Zielfunktion, was nicht mit unserer Problemstellung vereinbar ist.
Schließlich entscheiden wir uns für einen Ansatz mit sogenannten Radialen Basisfunktionen. Sie bringen alle gewünschten Eigenschaften mit sich. Nach der Definition wird die Existenz und Eindeutigkeit ausführlicher als in der einschlägigen Literatur gezeigt. Dies ermöglicht später, bei der Hinzunahme von zusätzlichen Interpolationspunkten viel Rechenzeit zu sparen, was aber keine neue Methode ist.
Das zweite Kapitel liefert eine Klassifizierung und Übersicht der wichtigsten bekannten Verfahren, die für die sequentielle Problemstellung entwickelt wurden. Alle Verfahren werden beschrieben und algorithmisch dargestellt.
Die simpelsten Methoden sind globale Optimierung von Approximations- bzw. Interpolationsfunktionen. Trotzdem geht hieraus das Verfahren CORS hervor, daß auf diesem Gebiet als eines der beiden besten gilt.

Um die weiteren Ansätze zu erklären, wird eine kurze Einführung in die Technik des Kriging gegeben. Dabei wird ein statistischer Zugang gewählt. Damit lassen sich die Verfahren DIRECT und EGO erklären, die zwar nicht mehr ganz neu sind, aber dennoch in der Literatur zu Vergleichszwecken herangezogen werden.
Abgeschlossen wird die Übersicht durch die sogenannten einstufigen Verfahren, zu denen die RBF Methode und das in Kapitel 4 vorgestellte neue Verfahren gehören.
In Kapitel drei werden zu Beginn das Newton-Verfahren und das Halley-Verfahren hergeleitet. Es wird gezeigt, daß das Halley-Verfahren von dritter Ordnung ist. Dies wird in der Literatur immer wieder erwähnt, jedoch nicht für mehr als eine Dimension bewiesen, zumindest konnte trotz intensiver Recherche kein Beweis für beliebige Dimension gefunden werden.
Schließlich wird die verwendete Herleitung verallgemeinert und damit das neue Verfahren $\mathcal{V}$ konstruiert. Es wird gezeigt, daß dieses neue Verfahren von vierter Ordnung ist.
Es wird überprüft, ob Halley-Verfahren und $\mathcal{V}$ überhaupt im Rahmen der vorliegenden Problemstellung durchgeführt werden können. Bei beiden Verfahren ist das Ergebnis positiv, wobei man mit der Durchführung von $\mathcal{V}$ die Möglichkeiten voll ausreizt.
Es folgt eine sehr ausführliche Testreihe samt Statistik an den üblichen Testfunktionen: Das Halley-Verfahren stellt sich als klarer Sieger im Vergleich zum Newton-Verfahren heraus. Dagegen hätte man trotz des erheblichen Aufwands, den man in $\mathcal{V}$ investieren müßte, keinen lohnenswerten Effekt im Vergleich zum Halley-Verfahren. Eine Testreihe dieser Größenordnung für das Halley-Verfahren konnte in der Literatur nicht gefunden werden. Dort hat man, wenn überhaupt, meist an einer Funktion und mit einem Startwert verglichen. Deshalb fiel die Entscheidung, das Halley-Verfahren als lokalen Sucher in das Gesamtverfahren zu integrieren.
Das vierte Kapitel ist das wichtigste dieser Arbeit: Es wird zunächst ein einstufiges Verfahren für die sequentielle Problemstellung hergeleitet und als Algorithmus beschrieben. Das Verfahren wurde in Matlab implementiert und auf die für die sequentielle Fragestellung gängigen Testfunktionen angewendet. Die Ergebnisse werden in einer Übersicht mit denen der gängigen Verfahren verglichen. Als bisher beste Verfahren gelten CORS und die beiden RBF Implementierungen. Sie haben beide den Nachteil, nur bei jeweils ungefähr einer Hälfte der Testbeispiele sehr gut abzuschneiden, während der Konkurrent bei der anderen Hälfte deutlich besser ist. Das neue Verfahren konnte für jede Testfunktion das bisherige Bestergebnis entweder übertreffen oder zumindest fast erreichen und ist somit konkurrenzfähig.

Aus dem sequentiellen Verfahren wird das parallele hergeleitet, es hat dieselbe Grundstruktur. Jedoch ist im Vergleich zum sequentiellen Verfahren die lokale Suche wesentlich aufwendiger: Es wird eine Heuristik entwickelt, die verhindern soll, daß lokale Suche immer wieder in den selben Punkten hängenbleibt. Das Verfahren wird am selben Testsatz wie die sequentielle Fassung getestet und schneidet sehr gut ab. Das globale Minimum wurde immer gefunden, auch wenn das Verfahren vorher in einem falschen Minimum gesucht hat. Leider können in der Literatur keine Vergleichsdaten gefunden werden. Es scheint allerdings sehr unwahrscheinlich, daß man die Ergebnisse in diesem Testsatz noch deutlich unterbieten kann.
Der Ausblick schließt diese Arbeit ab. Es werden resümiert und angedeutet, welche weiteren Fragestellungen Bestandteil zukünftiger Arbeit sein könnten.
Im Anhang finden sich die Definitionen der verwendeten Testfunktionen.

Kapitel 1

Interpolationstechniken

Interpolationstechniken sind ein grundlegender Bestandteil der meisten in der Literatur vorgestellten Verfahren, die für ähnliche Problemstellungen entwickelt wurden. Dies gilt auch für die in dieser Arbeit vorgestellte Methode. Es wird die Aufgabe auftreten, zu gegebenen $x_i \in \mathbb{R}^n$ und $y_i \in \mathbb{R}$ eine analytisch gegebene, mindestens zwei mal stetig differenzierbare Funktion Φ zu finden, die $\Phi(x_i) = y_i$ für alle vorliegenden x_i erfüllt. Die Mengen $\Xi = \{x_1, \ldots, x_m\}$ und $Y = \{y_1, \ldots, y_m\}$ seien dabei endlich. Je nach Kontext sind sie gelegentlich auch als Tupel aufzufassen. Was gemeint ist, wird an der entsprechenden Stelle klar. Jedes x_i darf nur einmal in Ξ vorhanden sein.
Die Bearbeitung unserer Problemstellung wird in jeder Iteration die geschickte Wahl neuer Auswertungsstellen fordern. Um möglichst frei in dieser Wahl zu sein, möchten wir an die Verteilung der x_i im $\mathbb{R}^n$ möglichst wenige Anforderungen stellen. So ist es z.B. wünschenswert, daß die Einhaltung bestimmter Gitterstrukturen o.Ä. nicht notwendig für die Existenz von Φ ist. Man spricht hier von sogenannten „scattered data“. Weiterhin werden wir fordern müssen, daß es möglichst wenig aufwendig ist, Ξ um einen oder einige wenige Knoten zu ergänzen und dann den erweiterten Datensatz nach dem bisher verwendeten Prinzip zu approximieren bzw. interpolieren. Als Folge dieser Forderungen scheiden einige offensichtliche und auch andere bekannte, speziellere Ansätze aus.

1.1 Polynominterpolation

Im Eindimensionalen ist die Polynominterpolation der wohl einfachste und bekannteste Ansatz. Sei dazu $\mathbb{P}_n^k$ die Menge aller Polynome vom Grad k, die von $\mathbb{R}^n$ nach $\mathbb{R}$ abbilden. Dann kann im Eindimensionalen jedes m-elementige Ξ mit

einem eindeutigen Polynom aus $\mathbb{P}_1^{m-1}$ interpoliert werden. Im Mehrdimensionalen ist die Lage wesentlich komplizierter. Die Lösbarkeit des Problems sieht man leicht ein: wähle dazu φ_i aus $\mathbb{P}_n^1$ so, daß φ_i in x_i die einzige Nullstelle auf Ξ besitzt. Endliches Ξ vorausgesetzt, ist die Existenz unendlich vieler solcher Funktionen für jedes x_i offensichtlich. Man überzeugt sich nun durch einfaches Nachrechnen davon, daß das Polynom

$$\Phi(x) = \sum_{i=1}^{m} y_i \prod_{\substack{j=1 \\ j \neq i}}^{m} \frac{\varphi_j(x)}{\varphi_j(x_i)}$$

aus $\mathbb{P}_n^{m-1}$ ist und Y auf Ξ interpoliert. Es existiert also immer eine Lösung. Schon am Ansatz erkennt man, daß sie nicht eindeutig ist, wenn man auf weitere Forderungen verzichtet. Weiterhin würde man auf diese Art den Polynomgrad unvernünftig in die Höhe treiben, es wäre keine sinnvolle Approximation mehr gegeben, weil Polynome von hohem Grad an den Rändern des Interpolationsgebietes stark oszillieren.
In der Literatur wurden eine ganze Reihe von Ansätzen vorgestellt, die zur Eindeutigkeit führen. Es wurde jeweils von keiner festen Polynombasis ausgegangen, sondern eine solche in Abhängigkeit von Ξ konstruiert. Siehe zum Beispiel [28], [5] oder [21]. Damit bekommt man jedoch nicht den immensen Grad des Interpolanden in den Griff.
Anders ist dies beim Ansatz von de Boor und Ron, den man in [12] und [13] findet: Auch dort wird abhängig von Ξ ein eindeutiges interpolierendes Polynom konstriuert. Es ist in einem gewissen Sinne von minimalem Grad. Numerische Experimente haben ergeben, daß bei höchstens m zufällig verteilten Daten im $\mathbb{R}^{10}$ folgende Polynomhöchstgrade beobachtet wurden:

m	1	11	66	286	1001	3003	8008	19448
Polynomhöchstgrad	0	1	2	3	4	5	6	7

Bei einer „moderaten“Anzahl von Stützstellen scheint dies also eine Möglichkeit zu sein. Untersucht man den in [13] vorgestellten Algorithmus jedoch genauer, so erkennt man, daß bei der Hinzunahme von Knoten die Ergebnisse der bisherigen Rechnung nicht wiederverwendet werden können. Die gesamte Konstruktion muß erneut durchgeführt werden. Also ist dies doch keine Alternative für uns, zumal eine einmalige Durchführung schon deutlich länger dauert als eine Interpolation mit radialen Basisfunktionen, die später vorgestellt werden. Die später entwickelten Verfahren würden über 100 mal länger laufen, was selbst in unserem Rahmen nicht mehr akzeptabel wäre.

Für einen guten Überblick über die Polynominterpolation und deren Geschichte seien [18] und [17] empfohlen.

1.2 Abschnittsweise Polynominterpolation

Eine beliebte Technik, die im Eindimensionalen auch für beliebig viele Knoten weniger wellige Interpolanden erzeugt, ist die stückweise Polynominterpolation [11]: Man schafft sich Abhilfe, indem das Gesamtintervall in Teilintervalle unterteilt wird. Auf jedem Teilintervall wird ein Polynom angesetzt, womit man den Höchstgrad in den Griff bekommt. Im Eindimensionalen zweckmäßig und verständlich, ist es zu kompliziert und für uns nicht sinnvoll, dies auf beliebige Dimension zu erweitern, wenn man von beliebig verteilten Knoten ausgeht. Um dies zu verstehen, untersuchen wir die 2-D-Ansätze von Powell und Sabin [32] bzw. von Clough und Tocher (z.B. [7]), die in der Literatur immer wieder genannt werden.
Beide gehen zunächst davon aus, daß die konvexe Hülle von Ξ in Dreieicke unterteilt wird. Eine Delaunay Triangulierung, für die es zahlreiche Implementierungen gibt, liefert uns in nicht pathologischen Fällen eine vernünftige Aufteilung des Gebietes [20].
Nun soll auf jedem Dreieck ein Polynom so angesetzt werden, daß insgesamt ein Interpoland entsteht, der an den Übergangsstellen möglichst glatt ist. In [42] wird gezeigt, daß der Mindestgrad der angesetzten Polynome mindestens $4m + 1$ sein muß und man auf den Ecken die Ableitungen bis zur Ordnung $2m$ vorschreiben muß, wenn der Gesamtinterpoland aus C^m sein soll. Dies ist nicht wünschenswert. Man kann jedoch den Grad überschaubar halten, indem man jedes Dreieck erneut unterteilt.
Genau so wird im Ansatz von Powell und Sabin [32] vorgegangen, wo mit Polynomen vom Höchsgrad 2 gearbeitet wird. Es stellt sich heraus, daß eine Unterteilung in 6 bzw. 12 Unterdreiecke sinnvoll ist, wie sie in Abbildung 1.1 gezeigt wird. Wir wollen grob Ansätz und Ergebnisse abklären, ohne zu sehr auf Details einzugehen.

Wir betrachten zuerst die Unterteilung in sechs Segmente. Auf jedem wird ein Polynom vom Grad zwei angenommen, es sind also 36 Parameter zu bestimmen. Die Gesamtfunktion, die sich aus den sechs Polynomen zusammensetzt, soll auf dem Dreieck ABC einmal stetig differenziebar sein. Das ergibt 27 Bedingungen. Neun weitere erhält man, wenn in A, B und C Funktions- und Gradientenwerte vorgegeben werden. Es stellt sich heraus, daß die 36 Parameter durch die genannten 36 Bedingungen immer eindeutig bestimmt sind, somit können auf A, B und C beliebige Funktions- und Gradientenwerte gefordert werden.

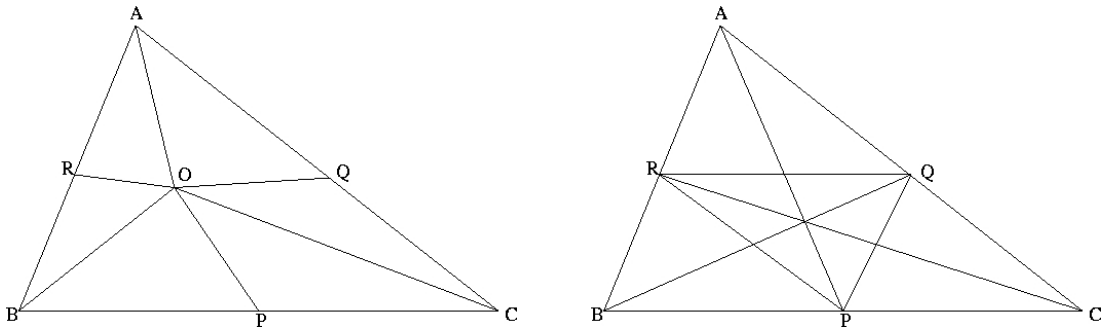


Abbildung 1.1: Unterteilung in 6 bzw. 12 Dreiecke nach Powell und Sabin

Dies ist nur ein Zwischenergebnis, man ist vielmehr an dem interessiert, was sich daraus für die Interpolation auf einer gegebenen Triangulierung ergibt. Entscheidend ist dabei, wo sich P, Q, R und O befinden. Legt man in jedem Dreieick O in den Umkreismittelpunkt, und sind P, Q und R jeweils die Seitenmittelpunkte, so ist der Interpoland auf der Gesamttriangulierung stetig differenzierbar. Es kann aber das Problem auftreten, daß der Umkreismittelpunkt außerhalb des Dreieicks liegt. In diesem Fall hilft man sich mit einer anderen Unterteilung.
P, Q und R halbieren jeweils die Seiten. Es wird dann in 12 Untersegmente aufgeteilt, siehe Abbildung 1.1, und schließlich analog zur Unterteilung von eben vorgegangen. Nach Erfüllen der C^1-Forderung auf ABC bleiben 12 Parameter mehr als Bedingungen. Deshalb schreibt man wieder Funktions- und Gradientenwerte in A, B bzw. C und zusätzlich die Normalenableitungen in P, Q und R vor. Dadurch wird erneut ein eindeutiger Interpoland erzeugt, der auch auf der ganzen Trinagulierung stetig differenzierbar ist.
Den C^1-Clough-Tocher Interpolanden [7] erhält man aus der Dreiteilung eines jeden Dreieicks, siehe Abbildung 1.2. Dort ist O der Schwerpunkt des Dreiecks. Auf jedem Mikrodreieck wird ein kubisches Polynom angesetzt, es sind also auf jedem Makrodreieick 30 Parameter zu bestimmen. Durch die Einhaltung der stetigen Differenzierbarkeit auf ABC ergeben sich 18 Forderungen. 12 weitere erhält man, indem man in A, B bzw. C Funktionswerte und Gradienten und in den Seitenmitten Normalenableitungen vorgibt. Das Ergebnis ist immer ein eindeutiger, auf der Gesamttriangulierung stetig differenzierbarer C^1-Interpoland.
Es bleibt zu klären, welche Vorgaben man für Gradienten bzw. Richtungsableitungen machen soll. Es gibt hier eine Fülle von Konstruktionsmöglichkeiten, wie z.B. die Berücksichtigung von Sekantensteigungen oder die Mimimierung gewisser Funktionale [24].
Sowohl die Taktik von Powell und Sabin als auch den C^1-Clough-Tocher Interpolanden kann man so erweitern, daß ein C^2-Interpoland entsteht [34]. Natürlich

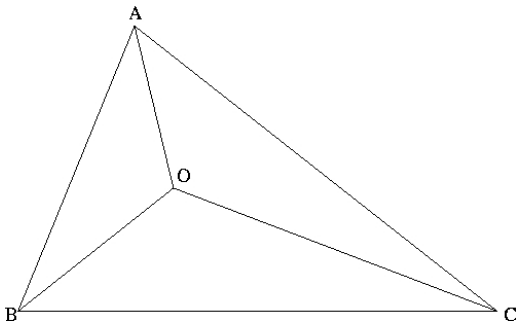


Abbildung 1.2: Unterteilung in 3 Dreiecke nach Clough und Tocher

bringt dies eine Erhöhung der Anzahl der Parameter und des Aufwandes mit sich. Wir wollen nicht zu sehr ins Detail gehen, da es sich nur um 2-D Interpolation handelt, wir aber etwa Dimension zehn anstreben.
Folglich ist es sinnvoll, nach Verallgemeinerungen der eben genannten Ansätze zu suchen. Ähnlich wie im bivariaten Fall würden ohne Unterteilung Polynome von hohem Grad auftreten. So benötigt man für einen trivariaten C^m-Interpolanden mindestens Grad $8m+1$ [41]. Es ist zu erwarten, daß sich dies für höhere Dimension fortsetzt. Also ist eine Unterteilung in Untersimplizes notwendig. Dies war schon im Zweidimensionalen nicht völlig trivial und man kann sich vorstellen, daß eine sinnvolle Fortsetzung im Dreidimensionalen nicht leicht ist. Dennoch wird dies in der Literatur erwähnt. So wird zum Beispiel in [39] ein n-dimensionaler C^1-Clough-Tocher-Interpoland angegeben. Dort wird jedes Makro-Simplex in $\frac{(n+1)!}{2}$ Mikro-Simplizes unterteilt. Der polynomiale Grad ist dort jeweils kubisch. Es ist klar, daß dies in unserer Anwendung viel zu viele Parameter mit sich bringt.
Weniger Mikro-Simplizes treten auf, wenn man dadurch aufteilt, daß der Schwerpunkt mit den Ecken des Simplex verbunden wird, wie es in [1] getan wird. Allerdings bezahlt man dies damit, daß der Polynomgrad unvernünftig hoch wird, was ja gerade vermieden werden sollte.
Der Ansatz von Powell und Sabin wurde in [40] verallgemeinert, allerdings trat schon im trivariaten Fall eine Aufteilung in 24 Untersimplizes auf, was keine Hoffnung für den beliebigdimensionalen Fall macht.
Zu all diesen Erschwernissen kommt hinzu, daß im Höherdimensionalen als Verallgemeinerung der Triangulierung eine Unterteilung in entsprechende Simplizes benötigt wird. Dies bereitet für $n \approx 10$ größte Schwierigkeiten [2]. Wir müssen also nach anderen Alternativen suchen.

1.3 Shepard Methoden

Für unsere Zwecke geeigneter scheint der aus der Meteorologie und Geologie entstandene ([9] und [8]) Shepard Ansatz zu sein. Er unterscheidet sich deutlich von dem, was wir bisher betrachtet haben. Von Vorteil wird sein, daß Aufwand und Durchführbarkeit von wachsender Dimension kaum negativ beeinflußt werden. Als interpolierende Funktion Φ wird kein Polynom, bzw. stückweises Polynom, sondern

$$\Phi(x) = \sum_{i=1}^{m} \omega_i(x) y_i$$

angesetzt, wobei

$$\omega_i(x) = \frac{\sigma_i(x)}{\sum_{j=1}^{m} \sigma_j(x)}$$

die sogenannten Gewichtsfunktionen sind und für gewöhnlich

$$\sigma_i(x) = \frac{1}{\|x - x_i\|^{\mu_i}}, \quad \mu_i > 0$$

gesetzt wird. Es werden also die y_i gewichtet: Je weiter man sich von x_i entfernt, desto weniger wird y_i berücksichtigt.
In Abbildung 1.3 wurde willkürlich $\Xi_B = \{1, 2, 3, 3.3, 3.6, 3.9, 4\}$ und $Y_B = \{0.6854, 1.5587, 4.3548, 0.0148, 4.3100, 4.2592, 2.5273\}$ gewählt. Die linke Graphik zeigt ω_3 für $\mu_i = 0.8$ bzw $\mu_i = 2$ (für $i = 1, \ldots, 7$), die rechte den daraus resultierenden Shepard-Interolanden. Die Knoten Ξ_b, Y_B werden in weiteren Beispielen wieder verwendet.

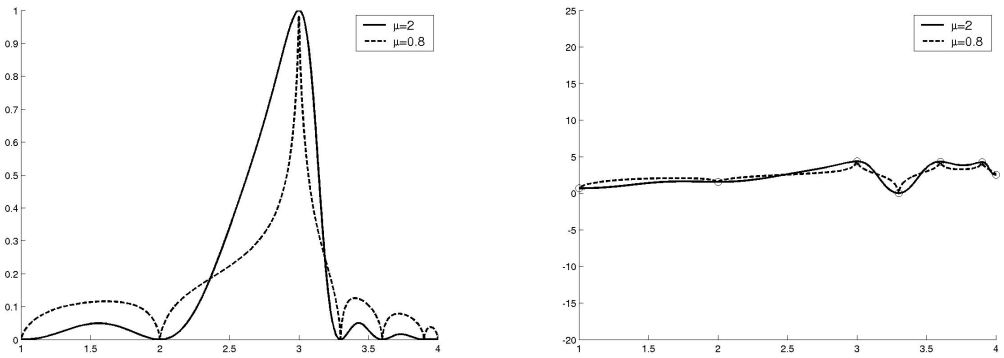


Abbildung 1.3: Shepard Gewichtsfunktionen und Interpolanden

Durch Nachrechnen, dies beinhaltet u.a. Grenzwertermittlung in den Stützstellen, erkennt man, daß Φ tatsächlich ein stetiger Interpoland ist. Allerdings deutet Abbildung 1.3 einen Nachteil an, der nicht zufällig im gewählten Beispiel, sondern immer auftritt und in [22] gezeigt wird: Für ein beliebiges μ_i gelte $0 < \mu_i \leq 1$. Dann ist Φ in x_i nicht differenzierbar. Haben wir dagegen für ein beliebiges μ_i die Bedingung $\mu_i > 1$, so gilt $\nabla\Phi(x_i) = 0$. Dies gilt für alle in Φ auftretenden μ_i. Wir haben also in allen Stützstellen entweder Knicke oder Gradientennullstellen. Beide Eigenschaften sind nicht wünschenswert: Warum sollte die eigentliche Funktion in den Stützstellen nicht differenzierbar sein bzw. der Gradient eine Nullstelle besitzen?
Dies kann durch modifizierte Ansätze [31] umgangen werden, jedoch muß man dann in den x_i Ableitungsinformationen, bzw. deren Annäherung beschaffen, was aber im Rahmen unserer Problemstellung nicht möglich ist.
Die gennanten Schwierigkeiten motivieren die Suche nach Alternativen. Man findet sie im Kalkül der Radialen Basisfunktionen.

1.4 Radiale Basisfunktionen

Auch hier wird abhängig von Ξ eine Basis von Funktionen bestimmt. Im Gegensatz zum Polynomansatz in [13] muß man keine aufwendige Rechnung durchführen, um die Basisfunktionen zu bestimmen. Sie zeichnen sich dadurch aus, nur von der Entfernung zu den Knoten abzuhängen. Wir wählen im Folgenden, wenn nicht explizit etwas Anderes genannt wird, immer die euklidische Norm.

1.4.1 Ansätze und Eigenschaften

Definition 1.1. (Radiale Basisfunktion) *Sei* $d_i(x) := \|x - x_i\|_2$*, wobei* $x \in \mathbb{R}^n$ *und* $\{x_1, \ldots, x_m\} = \Xi \subset \mathbb{R}^n$*. Wir nennen eine stetige Funktion* $\phi : \mathbb{R}_+ \to \mathbb{R}$ Radiale Basisfunktion*, falls* $\phi = \phi(d_i(x))$*.*

Aus einer geeigneten Linearkombination solcher Funktionen soll sich dann der gesuchte Interpoland zusammensetzen:

$$\Phi(x) = \sum_{j=1}^{m} \lambda_j \phi(d_j(x)), \tag{1.1}$$

wobei $\lambda_1, \ldots, \lambda_m \in \mathbb{R}$. Gebräuchliche Basisfunktionen ϕ sind

$$\phi(d) = e^{-\alpha d^2} \quad \text{Gauß'scher Ansatz, } \alpha > 0 \tag{1.2}$$
$$\phi(d) = \sqrt{d^2 + c^2} \quad \text{Multiquadrik, } c \in \mathbb{R} \tag{1.3}$$
$$\phi(d) = \sqrt{d^2 + c^2}^{-1} \quad \text{Inverse Multiquadrik, } c \in \mathbb{R}. \tag{1.4}$$

Die Interpolationsbedingung der Funktionswerte $y_i \in Y$ in den Stellen $x_i \in \Xi$ bedeutet knotenweise

$$\Phi(x_i) = \sum_{j=1}^{m} \lambda_j \phi(d_j(x_i)) = y_i$$

für $i = 1, \ldots, m$. Sie läßt sich kompakter schreiben als Lineares Gleichungssystem

$$P\lambda = Y, \tag{1.5}$$

wobei

$$P = \begin{pmatrix} \phi(0) & \phi(\|x_2 - x_1\|) & \cdots & \phi(\|x_m - x_1\|) \\ \phi(\|x_1 - x_2\|) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \phi(\|x_m - x_{m-1}\|) \\ \phi(\|x_1 - x_m\|) & \cdots & \phi(\|x_{m-1} - x_m\|) & \phi(0) \end{pmatrix}.$$

und natürlich $\lambda = (\lambda_1, \ldots, \lambda_m)^T$. In [4] wird die Invertierbarkeit von P für die Ansätze (1.2), (1.3) und (1.4) für beliebige endliche $\Xi \subset \mathbb{R}^n$ gezeigt. Damit erlauben diese Ansätze immer eine eindeutige Interpolation.
(1.5) dient nicht nur theoretischen Zwecken, sondern wird in der Praxis tatsächlich gelöst. Unsere Problemstellung erlaubt uns auch bei größeren Systemen (≈ 5000 bis 10000 Variablen) ein direktes Lösen. Bei Testinterpolationen dieser Größenordnung im $\mathbb{R}^{10}$ traten keine Probleme mit schlecht konditionierten Matrizen auf. Zur Illustration dient Abbildung 1.4. Dort werden die drei Basisfunktionen aus (1.2), (1.3) und (1.4) an sich sowie die Interpolation des Datensatzes aus Abschnitt 1.3 mit dem Interpolanden (1.1), in den jeweils die Basisfunktionen (1.2), (1.3) und (1.4) eingehen, gezeigt. Es wurde jeweils $\alpha = c = 1$ gewählt. Die richtige Wahl der Parameter ist besonders für die Ansätze (1.2) und (1.4) mitentscheidend für die Brauchbarkeit des Resultates. Dies ist kein gravierender Nachteil, weil es auch parameterfreie Verfahren gibt. Diese kommen zudem ohne die auffallend und scheinbar unnötig großen Amplituden aus, die in unserem Beispiel für $x \approx 1.5$ und $x \approx 2.5$ auftreten. Dieser Vorteil läßt sich sogar präzise formulieren und kristallisiert sich als Eigenschaft der folgenden Herangehensweise heraus

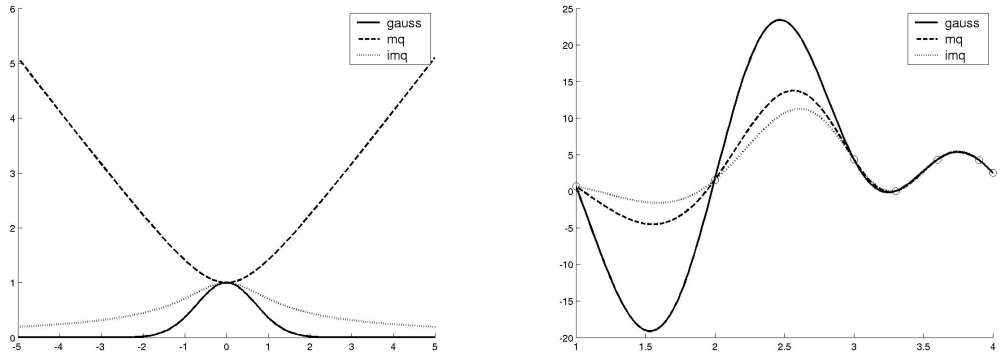


Abbildung 1.4: Gauß'scher Ansatz und Multiquadriken

((1.12) und Satz 1.5). Um Existenz und Eindeutigkeit solcher Interpolanden sicherstellen zu können, werden sie um ein Polynom ergänzt. Es seien also die ab nun betrachteten Φ von der Form

$$\Phi(x) = \sum_{i=1}^{m} \lambda_i \phi_i(d_i(x)) + p(x), \tag{1.6}$$

mit $p \in \mathbb{P}_n^{k-1}$. Für das Weitere müssen zwei Begriffe eingeführt werden.

Definition 1.2. (Bedingt Positiv Definit) *Eine Funktion $F : \mathbb{R}^n \to \mathbb{R}$ heißt* bedingt positiv definit von der Ordnung k, *falls für alle endlichen $\{x_1, \ldots, x_m\} \subset \mathbb{R}^n$ und alle $\{\lambda_1, \ldots, \lambda_m\}$ mit $\sum_{i=1}^m \lambda_i q(x_i) = 0$ für alle $q \in \mathbb{P}_n^{k-1}$*

$$\sum_{i,j=1}^{m} \lambda_i \lambda_j F(x_i - x_j) \geq 0 \tag{1.7}$$

gilt. Wir nennen F strikt bedingt positiv definit, *falls in (1.7) sogar „ $>$ “ gilt.*

Definition 1.3. (Unisolvent) *Eine endliche Menge $\Xi \subset \mathbb{R}^n$ heißt* unisolvent *für $\mathbb{P}_n^{k-1}$, falls $p(x_i) = 0$ für alle $x_i \in \Xi$ impliziert, daß $p \equiv 0$.*

So ist z.B. für lineare Polynome von $\mathbb{R}^n$ nach $\mathbb{R}$ schon jede $n+1$-elementige Menge, die nicht auf einer $n-1$-dimensionalen Hyperebene liegt, unisolvent (Im Zweidimensionalen also 3 Punkte, die nicht auf einer Linie liegen).
Diese beiden Begriffen führen zu einer Existenz- und Eindeutigkeitsaussage. Der Multiindex $\alpha \in \mathbb{N}^n$ wird in der gängigen Weise verwendet.

Satz 1.4. *Sei $F : \mathbb{R}^n \to \mathbb{R}$ strikt bedingt positiv definit von der Ordnung k und $\{x_1, \ldots, x_m\}$ unisolvent für $\mathbb{P}_n^{k-1}$. Dann gibt es für beliebige auf $\{x_1, \ldots, x_m\}$ vorgegebene Werte genau einen Interpolanden der Form*

$$\Phi(x) = \sum_{i=1}^{m} \lambda_i F(x - x_i) + p(x) \quad \text{mit} \quad p \in \mathbb{P}_n^{k-1}$$

der zusätzlich die Bedingung

$$0 = \sum_{i=1}^{m} \lambda_i x_i^\alpha \quad \textit{für} \quad |\alpha| < k \tag{1.8}$$

erfüllt.

Beweis: Eine sehr knappe Fassung bietet [4]. Um bestimmte Vorgehensweisen bei der späteren Umsetzung besser erklären zu können, wird hier ausführlicher verfahren.
Seien $\lambda = (\lambda_1, \ldots, \lambda_m)^T$ die Vorfaktoren der ϕ_i in (1.6) und $\beta = (\beta_0, \ldots, \beta_{\tilde{n}})^T$ die Koeffizienten des dort erwähnten Polynoms p, also

$$p(t) = \beta_0 + \beta_1 t_1 + \cdots + \beta_n t_n + \beta_{n+1} t_1^2 + \beta_{n+2} t_1 t_2 + \cdots + \beta_{\tilde{n}} t_n^{k-1}$$

Aus der Interpolationsbedingung und der Einhaltung von (1.8) ergibt sich das Lineare Gleichungssystem

$$\begin{pmatrix} P & Q \\ Q^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \beta \end{pmatrix} = \begin{pmatrix} Y \\ 0 \end{pmatrix}, \tag{1.9}$$

wobei

$$P = \begin{pmatrix} F(0) & F(x_1 - x_2) & \cdots & F(x_1 - x_m) \\ F(x_2 - x_1) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & F(x_{m-1} - x_m) \\ F(x_m - x_1) & \cdots & F(x_m - x_{m-1}) & F(0) \end{pmatrix},$$

$$Q = \left(\begin{array}{c|ccc|ccccc|c} 1 & x_1^1 & \cdots & x_1^n & (x_1^1)^2 & x_1^1 x_1^2 & \cdots & (x_1^n)^2 & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \\ 1 & x_m^1 & \cdots & x_m^n & (x_m^1)^2 & x_m^1 x_m^2 & \cdots & (x_m^n)^2 & \cdots \end{array}\right.$$

$$\left.\begin{array}{c|ccc} \cdots & (x_1^1)^{k-1} & \cdots & (x_1^n)^{k-1} \\ & \vdots & & \vdots \\ \cdots & (x_m^1)^{k-1} & \cdots & (x_m^n)^{k-1} \end{array}\right).$$

Wir haben also zu zeigen, daß die Gesamtmatrix in (1.9) invertierbar ist. Es darf demnach kein $(\tilde{\lambda}, \tilde{\beta})^T \neq 0$ geben mit

$$\begin{pmatrix} P & Q \\ Q^T & 0 \end{pmatrix} \begin{pmatrix} \tilde{\lambda} \\ \tilde{\beta} \end{pmatrix} = 0 \tag{1.10}$$

Man unterscheidet zwei Fälle:

1) (1.10) werde gelöst, wobei $\tilde{\lambda} = 0$ gelte. Daraus würde sofort $Q\tilde{\beta} = 0$ folgen. Dies wäre nichts Anderes, als daß das durch die Koeffizienten aus $\tilde{\beta}$ definierte Polynom aus $\mathbb{P}_n^{k-1}$ auf Ξ verschwinden würde. Da Ξ aber unisolvent ist für $\mathbb{P}_n^{k-1}$, müßte dieses Polynom konstant null sein. Also würde auch $\tilde{\beta} = 0$ gelten. Auf diese Art kann also kein $(\tilde{\lambda}, \tilde{\beta})^T \neq 0$, das (1.10) genügt, konstruiert werden.

2) (1.10) werde gelöst, wobei $\tilde{\lambda} \neq 0$ gelte. Das würde natürlich $P\tilde{\lambda} + Q\tilde{\beta} = 0$ nach sich ziehen. Multiplikation von links mit $\tilde{\lambda}^T$ müßte dann auch null ergeben. Wir erhalten aber

$$\tilde{\lambda}^T(P\tilde{\lambda} + Q\tilde{\beta}) = \tilde{\lambda}^T P\tilde{\lambda} + \underbrace{\tilde{\lambda}^T Q\tilde{\beta}}_{=0 \text{ wegen (1.10)}} = \sum_{x_i, x_j \in \Xi} \lambda_{x_i}\lambda_{x_j} F(x_i - x_j) \overset{F \text{ s.b.p.d.}}{>} 0.$$

Solch ein Skalarprodukt wird natürlich nur dann ungleich null, wenn beide Faktoren ungleich null sind. Also kann $P\tilde{\lambda} + Q\tilde{\beta}$ nicht null sein. Folglich gibt es kein $(\tilde{\lambda}, \tilde{\beta})^T \neq 0$ mit $\tilde{\lambda} \neq 0$ das (1.10) erfüllt.

Es gibt also kein (1.10) erfüllendes $(\tilde{\lambda}, \tilde{\beta})^T \neq 0$, was die Invertierbarkeit und damit die Aussage beweist. □

Wählen wir nun ein ganzzahliges k mit $k > \frac{n}{2}$ und betrachten die Basisfunktionen

$$\phi(d) = \begin{cases} d^{2k-n} \log d & \text{falls } 2k-n \text{ gerade} \\ d^{2k-n} & \text{falls } 2k-n \text{ ungerade.} \end{cases} \tag{1.11}$$

Diese Ansätze liefern strikt bedingt positiv definite Funktionen [4] und damit einen eindeutigen Interpolanden der Form (1.6). Dieser zeichnet sich bezüglich der Glattheit in einem bestimmten Sinn aus. Zur Erläuterung verwenden wir den Multiindex $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}_0^n$, für den wir

$$|\alpha| := \alpha_1 + \cdots + \alpha_n \quad \text{und} \quad \alpha! := \alpha_1!\alpha_2! \cdot \cdots \cdot \alpha_n!$$

und damit für eine entsprechend oft differenzierbare Funktion

$$D^\alpha f := \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}}$$

festlegen. Damit definieren wir die Seminorm

$$\|f\|_k^* := \int_{\mathbb{R}^n} \sum_{|\alpha|=k} \frac{k!}{\alpha!} \left(D^\alpha f(x)\right)^2 dx. \tag{1.12}$$

Für sie gilt in Verbindung mit (1.11) das interessante Resultat

Satz 1.5. *Sei $\{x_1, \dots, x_m\} \subset \mathbb{R}^n$ endlich und enthalte eine für $\mathbb{P}_n^{k-1}$ unisolvente Teilmenge. Dann gibt es für beliebige auf $\{x_1, \dots, x_m\}$ vorgegebene Werte genau einen Interpolanden $s(x)$ der Form (1.6) mit Basisfunktionen nach (1.11), für den $\sum_{i=1}^m \lambda_i q(x_i) = 0$ für alle $q \in \mathbb{P}_n^{k-1}$ gilt. Weiterhin minimiert s die Seminorm (1.12).*

Beweis: Siehe [15]. □

Im Falle von $k = 2$ und $n = 1$ spricht man vom kubischen Ansatz, man erhält dann einen Interpolanden der Form

$$\Phi(x) = \sum_{i=1}^m \lambda_i |x - x_i|^3 + ax + b.$$

mit $a, b \in \mathbb{R}$. Für $k = 2$ und $n = 3$ spricht man vom linearen Ansatz. Man erhält

$$\Phi(x) = \sum_{i=1}^m \lambda_i \|x - x_i\| + ax^T + b$$

mit $a \in \mathbb{R}^3$ und $b \in \mathbb{R}$. Für $k = n = 2$ spricht man vom Thin Plate Spline.

$$\Phi(x) = \sum_{i=1}^m \lambda_i \|x - x_i\|^2 \log\left(\|x - x_i\|\right) + ax^T + b,$$

wobei $a \in \mathbb{R}^2$ und $b \in \mathbb{R}$. Letzterer Begriff wurde nicht ohne Motivation gewählt. In diesem Fall mißt

$$\int_{\mathbb{R}^2} \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2$$

die Biegeenergie einer dünnen, unendlich ausgedehnten elastischen Platte, die in den Interpolationspunkten befestigt ist.
Wählen wir $n = 1$ und $k = 2$, so erhalten wir den bekannten Natürlichen Kubischen Spline. Um sich dies klar zu machen, sei in diesem speziellen Fall $\Xi = \{x_1, \ldots, x_m\}$ mit $x_1 < \cdots < x_m$. Offensichtlich ist (1.6) zwischen den Knotenpunkten jeweils ein Polynom vom Grad drei und minimiert

$$\int_{\mathbb{R}} \left(\frac{\partial^2 f}{\partial x} \right)^2 dx = \int_{x_1}^{x_m} f''(x)^2 dx.$$

Es bedarf einer kleinen Rechnung, um die Gültigkeit dieser Gleichung einzusehen. Betrachten wir den Interpolanden für $x < x_1$ und $x > x_m$, so gilt dort stets $\phi_i = -(x - x_i)^3$ (bzw. $\phi_i = (x - x_i)^3$) und damit

$$\begin{aligned} \sum_i \lambda_i \phi_i &= -\sum_i \lambda_i (x - x_i)^3 = -\sum_i \lambda_i \left(x^3 - x^2 x_i + x x_i^2 - x_i^3 \right) \\ &= -x^3 \sum_i \lambda_i + x^2 \sum_i \lambda_i x_i - x \sum_i \lambda_i x_i^2 + \sum_i \lambda_i x_i^3 \\ &\overset{(1.8)}{=} 0 \end{aligned}$$

Da das lineare Polynom im Interpolanden auch keinen Beitrag zur Krümmung leistet, darf man das Integral auf $[x_1, x_m]$ einschränken. Mit der Nullsume in (1.8) begründet man auch $f''(x_1) = f''(x_m) = 0$. Somit liegt der interpolierende natürliche Spline vor. Der Ansatz (1.11) ist also eine Verallgemeinerung dieser beliebten Methode auf beliebigdimensionale Räume. Abbildung 1.5 dient der Illustration. Es wurde derselbe Datensatz wie zu Erzeugung von Abbildung (1.4) verwendet. Der Betrachter hat im Vergleich zu dort den Eindruck, daß hier die vorliegenden Daten treffender interpretiert werden.

1.4.2 Hinzunehmen von Stützstellen

Für uns wird es sehr interessant sein, möglichst wenig Rechenzeit zu investieren, wenn die schon interpolierte Menge Ξ um einen zusätzlichen Knoten x^z und einen dort vorgegebene Funktionswerte y^z ergänzt wird. Aus dem schon bekannten Interpolanden Φ soll nun möglichst effizient der Interpoland $\tilde{\Phi}$ der Form (1.6) berechnet werden, der auf der Menge $\Xi \cup x^z$ interpoliert. Bei der Berechnung von Φ wurde die Lösung x_Φ eines Linearen Gleichungssystems $A_\Phi x_\Phi = b_\Phi$ bestimmt. Wir drücken dies symbolisch dadurch aus, daß wir im Folgenden A_Φ^{-1} schreiben, was in der Anwendung allerdings immer eine Rücksubstitution im Rahmen der

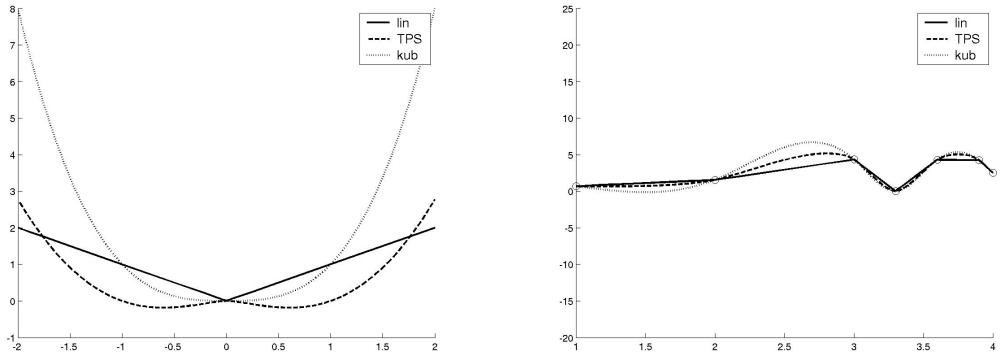


Abbildung 1.5: Linearer und Kubischer Ansatz, Thin Plate Spline

bereits berechneten LR-Zerlegung von A_Φ bedeutet. Die Hinzunahme von x^z führt zu dem Linearen Gleichungssystem

$$\begin{pmatrix} \phi(0) & Q_{\tilde{\Phi}} \\ Q_{\tilde{\Phi}}^T & A_\Phi \end{pmatrix} \begin{pmatrix} \tilde{\lambda} \\ \tilde{\beta} \end{pmatrix} = \begin{pmatrix} y^z \\ Y \\ 0 \end{pmatrix} \tag{1.13}$$

$Q_{\tilde{\Phi}}$ ist in diesem Fall folgendermaßen strukturiert:

$$\begin{aligned} Q_{\tilde{\Phi}} = \big(\; & \phi(\|x^z - x_1\|) \quad \cdots \quad \phi(\|x^z - x_m\|) \;\big|\; 1 \;\big|\; x_1^z \quad \cdots \quad x_n^z \;\big| \\ & (x_1^z)^2 \quad x_1^z x_2^z \quad \cdots \quad (x_n^z)^2 \;\big|\; \cdots \;\big|\; (x_1^z)^{k-1} \quad \cdots \quad (x_n^z)^{k-1} \;\big) . \end{aligned} \tag{1.14}$$

Die Vektoren sind

$$\tilde{\lambda} = (\tilde{\lambda}_1, \ldots, \tilde{\lambda}_{m+1})^T \quad \text{und} \quad \tilde{\beta} = (\tilde{\beta}_0, \ldots, \tilde{\beta}_{\tilde{n}})^T$$

Wir gehen nun vom bereits vorhandenem A_Φ^{-1} aus. Wir schreiben (1.13) etwas um

$$\begin{pmatrix} \phi(0) & Q_{\tilde{\Phi}} \\ Q_{\tilde{\Phi}}^T & A_\Phi \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \end{pmatrix} = \begin{pmatrix} y^z \\ v_1 \end{pmatrix}, \tag{1.15}$$

mit $u_0 \in \mathbb{R}$ und $u_1, v_1 \in \mathbb{R}^{m+\tilde{n}}$. Die zweite Zeile kann man umformen zu

$$u_1 = A_\Phi^{-1}(v_1 - Q_{\tilde{\Phi}}^T u_0), \tag{1.16}$$

und Einsetzen in die erste Zeile liefert nach Umformung

$$u_0 = \frac{y^z - Q_{\tilde{\Phi}} A_{\Phi}^{-1} v_1}{\phi(0) - Q_{\tilde{\Phi}} A_{\Phi}^{-1} Q_{\tilde{\Phi}}^T}. \tag{1.17}$$

Wir fahren dann fort mit

$$\tilde{\lambda} = (u_0, u_1^1, \ldots, u_1^m) \text{ und } \tilde{\beta} = (u_1^{m+1}, \ldots, u_1^{m+\tilde{n}}).$$

Wir erfassen nun den Rechenaufwand der Teilschritte von (1.16) und (1.17) in tabellarischer Form. Daraus erschließen wir die Gesamtzahl von Operationen der Form $d = a + bc$ bzw. $d = a + b/c$, die wir zur Lösung von (1.15) benötigen.

$\underline{u_1}$:

Art der Rechnung	Eingehende Größen	Anzahl Operationen
Multiplikation	$Q_{\tilde{\Phi}}^T$, u_0	m
Subtraktion	v_1, $Q_{\tilde{\Phi}}^T u_0$	m
Rücksubstitution	A_{Φ}^{-1}, $(v_1 - Q_{\tilde{\Phi}}^T u_0)$	m^2

$$\Rightarrow \#\text{Operationen} = m^2 + 2m$$

$\underline{u_0}$:

Art der Rechnung	Eingehende Größen	Anzahl Operationen
Rücksubstitution	A_{Φ}^{-1}, $Q_{\tilde{\Phi}}^T$	m^2
Multiplikation	$Q_{\tilde{\Phi}}$, $A_{\Phi}^{-1} Q_{\tilde{\Phi}}^T$	m
Subtraktion	$\phi(0)$, $Q_{\tilde{\Phi}} A_{\Phi}^{-1} Q_{\tilde{\Phi}}^T$	1
Rücksubstitution	A_{Φ}^{-1}, v_1	m^2
Multiplikation	$Q_{\tilde{\Phi}}$, $A_{\Phi}^{-1} v_1$	m
Subtraktion	y^z, $Q_{\tilde{\Phi}} A_{\Phi}^{-1} v_1$	1
Division	$y^z - Q_{\tilde{\Phi}} A_{\Phi}^{-1} v_1$, $\phi(0) - Q_{\tilde{\Phi}} A_{\Phi}^{-1} Q_{\tilde{\Phi}}^T$	1

$$\Rightarrow \#\text{Operationen} = 2m^2 + 2m + 3$$

Damit benötigt man zum Auflösen von (1.13) von ca. $3m^2 + 4m + 3$, im wesentlichen also ca. $3m^2$ Operationen. Dies ist eine ganz erhebliche Ersparnis gegenüber

einer vollständigen LR-Zerlegung der Gesamtmatrix, die ca. $\frac{1}{3}m^3$ Operationen kosten würde.

Anmerkung: Eine völlig analoge Betrachtung läßt sich für eine Erweiterung um beliebig viele, z.B. $\tilde{m}$, Knoten durchführen. Es würde dann ein Aufwand von

$$m^2(\tilde{m}+2)+m(\tilde{m}+1)^2+\frac{1}{3}\tilde{m}+2\tilde{m}^2+\tilde{m}$$

Operationen enstehen. Dies ist im wesentlichen $m^2(\tilde{m}+2)$ und bei $\tilde{m} \ll m$ noch immer deutlich günstiger als die etwa $\frac{m^3}{3}$ Operationen die eine vollständige Zerlegung kosten würde.

Ein kurzes Rechenbeispiel deutet an, wie viel Rechenzeit mit dieser Methode eingespart werden kann.

Beispiel 1.1. (Numerisches Experiment) Wir wählen m gleichverteilte Punkte $x_1,\ldots,x_m$ auf $[0,1]^2$, dies geschieht mir dem `MATLAB` Befehl `rand`.Zu jedem dieser x_i wählen wir, wieder mit `rand`, ein y_i. Wir interpolieren nach dem kubischen Ansatz aus (1.11) was T_m Sekunden dauert.
Nun wird zufällig ein x^z aus $[0,1]^2$ und ebenso zufällig ein zugehöriges y^z aus $[0,1]$ gewählt. Wie eben beschrieben, wird keine LR-Zerlegung der neuen Gesamtmatrix berechnet, sondern die alte Zerlegung verwendet. Die neue Interpolation wird dann nach T_1 Sekunden berechnet.
Weiterhin werden zusätzliche 15 gleichverteilte $\tilde{x}_1,\ldots,\tilde{x}_{15}$ auf $[0,1]^2$ und zugehörige zufällige $\tilde{y}_i$ gewählt. Erneut wird keine LR-Zerlegung der neuen Gesamtmatrix berechnet, sondern die alte Zerlegung verwendet. Die neue Interpolation wird dann nach T_{15} Sekunden berechnet.
Es ergaben sich bei zehnfachem Durchführen des Experiments auf dem selben Rechner unter den selben Bedingungen folgende Mittelwerte ohne nennenswerte Abweichungen:

m	1000	2000	4000
T_m	2.9	16.4	101
T_{15}	0.4	1.4	5.5
T_1	0.3	1.0	4.3

Offenbar lohnt es sich mit steigender Knotenzahl immer mehr, die einmal gemachte Zerlegung wiederzuverwenden.

Kapitel 2

Bekannte Verfahren zur Optimierung teurer Funktionen

In Kapitel 1 wurde der Ansatz mit Radialen Basisfunktionen am ausführlichsten behandelt. Der Grund dafür ist, daß die in dieser Arbeit hergeleiteten Optimierungserfahren darauf aufbauen. Doch auch viele der bekannten Optimierungsmethoden enthalten derartige Bausteine. Wir wollen uns nun also mit den bisher entwickelten Ansätzen zur Optimierung extrem teurer Funktionen auseinandersetzen.
Dazu wird ein kurzer Überblick über die in der Literatur erwähnten Verfahren, die für die sequentielle Problemstellung entwickelt wurden, gegeben. Eine gute Klassifikation gibt [27]. Im Folgenden orientieren wir uns daran und ordnen die erfolgreichsten Methoden samt ihrer wichtigsten Eigenschaften ein. Dies ermöglicht auch, die in Kapitel 4 entwickelten Verfahren vernünftig beurteilen zu können.

2.1 Minimierung von Ausgleichsflächen

Man geht sicherlich am einfachsten vor, wenn der Datensatz $\Xi = \{x_1, \ldots, x_m\}$, $Y = \{y_1, \ldots, y_m\}$ zunächst durch eine Approximation Φ wiedergegeben wird. Ein einfacher Ansatz für Φ könnte z.B. die reelle Quadratische Form

$$\Phi := x^T D x + b^T x + c \tag{2.1}$$

sein, D ist eine Diagonalmatrix. Die Parameter D, b und c werden definiert z.B. durch die Bedingung

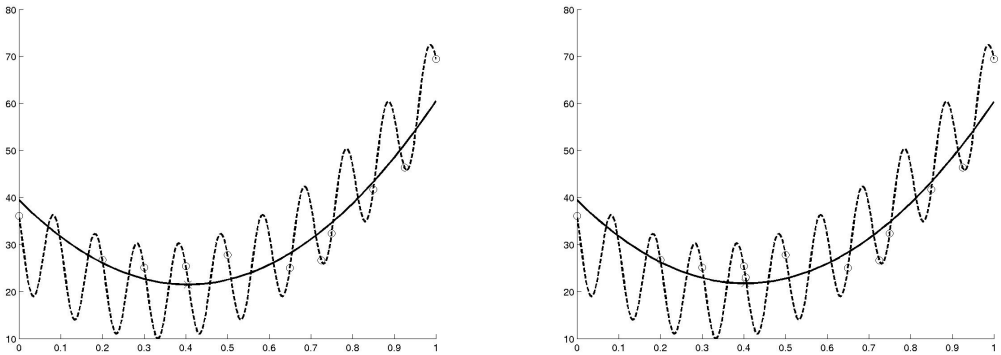
$$\min_{D,b,c} \sum_{i=1}^{m} (y_i - \Phi(x_i))^2 . \tag{2.2}$$

Nach Festlegung eines geeigneten Abbruchkriteriums lautet der Algorithmus zur Minimierung der Zielfunktion f dann:

Algorithmus 1 Minimierung Ausgleichsfläche
Input: Ξ, Y, f, n_{max} **Output:** x

1: Definiere $\Xi_0 := \Xi$ und $Y_0 := Y$
2: i:=0
3: **while** $i < n_{max}$ **do**
4: Bestimme Approximation Φ zu Ξ_i und Y_i
5: Finde globales Minimum x von Φ
6: Setze $\Xi_{i+1} := \Xi_i \cup x$ und $Y_{i+1} := Y_i \cup f(x)$
7: i:=i+1
8: **end while**

Algorithmus 1 wird am Beispiel der eindimensionalen Rastrigin Funktion durchgeführt, die Approximation wird nach (2.1) und (2.2) gewählt.
Man erkennt sofort den Nachteil, daß Konvergenz gegen einen nichtstationären Punkt auftreten kann. Siehe hierzu Abbildung 2.1, dort werden die Iterationen 1,2,3 und 1550 gezeigt. Diese Methode kann also höchstens untergeordneter Bestandteil eines ausgefeilteren Ansatzes sein.
Anmerkung In Zeile 4 des Algorithmus wird gefordert: „Finde globales Minimum x von Φ“. Dies an sich ist schon ein globales Minimierungsproblem und keineswegs einfach. Das gilt auch für die folgenden Algorithmen, in denen ähnliche Befehlszeilen auftreten. Dort ist die Situation sogar noch komplizierter, da in diesen Fällen keine quadratische Approximation, sondern sogar eine Interpolation mit komplizierteren Funktionen stattfindet. Die Auswertung von Φ ist nun aber vergleichsweise billig.



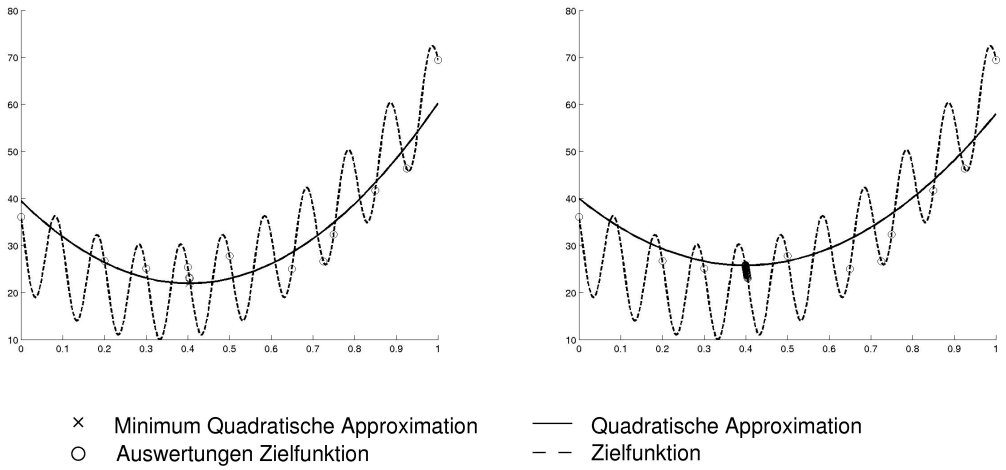


Abbildung 2.1: Minimierung der Zielfunktionsapproximation

Auch die in Kapitel 4 entwickelten Methoden verlangen die Lösung derartiger globaler Optimierungsprobleme. Wie dies implementiert wurde, ist in Abschnitt 4.3 beschrieben.

2.2 Minimierung von Interpolanden

Eine Aufwertung erfährt die eben beschriebene Technik, wenn Approximation durch Interpolation ersetzt wird. Wir gelangen dann zu einer Verfahrensvorschrift, die Algorithmus 1 bis auf Zeile 4 gleicht. Anstatt einer Approximationsfunktion wird nun eine Interpolationsfunktion minimiert. n_{max} ist die Anzahl der Schritte. Wir erhalten nun

Algorithmus 2 Minimierung Interpoland
Input: Ξ, Y, f, n_{max} **Output:** x

1: Definiere $\Xi_0 := \Xi$ und $Y_0 := Y$
2: i:=0
3: **while** $i < n_{max}$ **do**
4: Bestimme Interpoland Φ zu Ξ_i und Y_i
5: Finde globales Minimum x von Φ
6: Setze $\Xi_{i+1} := \Xi_i \cup x$ und $Y_{i+1} := Y_i \cup f(x)$
7: i:=i+1
8: **end while**

Die Hauptschwierigkeit hier ist es, im Mehrdimensionalen einen geeigneten Inter-

polanden Φ zu finden, was schon in Kapitel 1 diskutiert wurde. Auch die globale Minimierung von Φ ist durchaus nichttrivial. Im Eindimensionalen ist die Lage noch recht übersichtlich: Hier ist durch den natürlichen kubischen Spline eine vernünftige Interpolation für beliebiges Ξ möglich. Die anschließende Minimierung ist einfach. Abbildung 2.2 zeigt die Iterationen 1,2,3 und 25 von Algorithmus 2. Es wurde mit dem natürlichen kubischen Spline interpoliert und die selbe Startmenge wie im vorhergehenden Beispiel verwendet. Die Iteration konvergiert gegen ein lokales Minimum, was aber nicht immer so sein muß. Mögliche Abwandlungen, die zu lokalen Suchern führen, findet man in [27].
Erstaunlicherweise hat diese Grundidee in der globalen Optimierung teurer Funktionen vor kurzem zu einem der erfolgreichsten Verfahren geführt, CORS-RBF (Constrained Optimization using Response Surfaces (i.e. Radial Basis Functions)). Dieser in [33] vorgestellte Ansatz unterscheidet sich von Algorithmus 2 nur dadurch, daß der Interpoland Φ bezüglich der Nebenbedingung

$$\|x - x_j\| \geq \beta_i \Delta_i, \quad \forall x_j \in \Xi_i$$

mit

$$\Delta_i = \max_{\tilde{x} \in D} \min_{x_j \in \Xi} \|\tilde{x} - x_j\|$$

global minimiert wird. Im Laufe des Verfahren durchläuft β_i verschiedene Zyklen, woraus sich die beiden Versionen CORS-RBF (SP1), Zyklus=$< 0.95, 0.25, 0.05, 0.03, 0 >$, und CORS-RBF (SP2), Zyklus=$< 0.9, 0.75, 0.25, 0.05, 0.03, 0 >$, ergeben. Es wird also in jedem Zyklus zu Beginn mehr global und gegen Ende immer lokaler gesucht.
Trotz seines einfachen Aufbaus beeindruckt dieses Verfahren durch seine Ergebnisse in der Praxis und ist zudem noch für stetige Zielfunktionen global konvergent.

2.3 Kriging-Methoden

Mit dem Konzept der Radialen Basisfunktionen in Verbindung steht das sogenannte Kriging, welches statistisch interpretiert werden kann. Eine entsprechende Herleitung findet man in [35]. Wir wählen eine intuitivere Variante, die ausführlich in [26] und [27] beschrieben wird.
Ausgehend von der Aufgabe, an einer Stelle x den unbekannten Zielfunktionswert $f(x)$ zu schätzen bzw. vorherzusagen, werden die in $\Xi = \{x_1, \ldots, x_m\}$ gegebenen

Zielfunktionswerte $Y = \{y_1, \ldots, y_m\}$ aufgefaßt als Realisierungen der normalverteilten Zufallsvariablen $Z(x_1), \ldots, Z(x_m)$, die alle den Erwartungswert μ und die Varianz σ^2 besitzen.

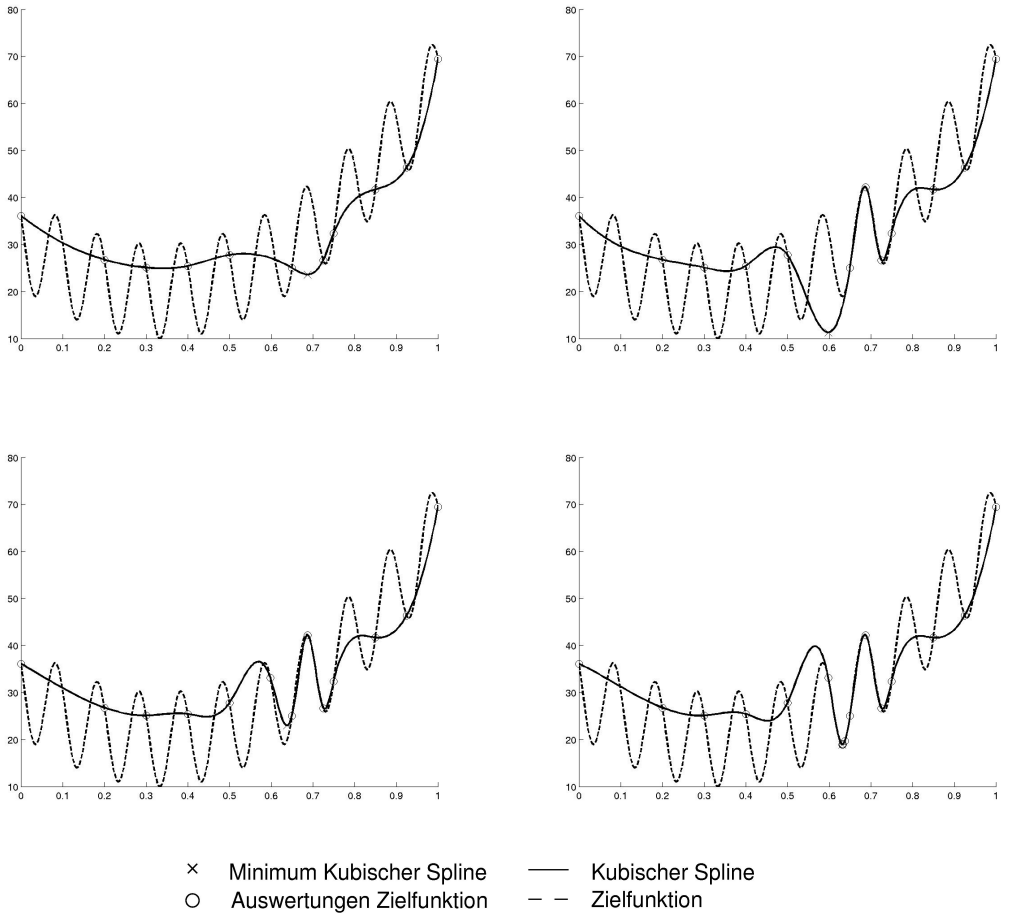


Abbildung 2.2: Minimierung der Zielfunktionsinterpolation

Die Korrelation von $Z(x_i)$ und $Z(x_j)$ sei durch

$$R_{i,j} := \mathrm{Corr}(Z(x_i), Z(x_j)) := \phi_i(x_j), \quad \phi_i(x) := \exp\left(-\sum_{l=1}^{n} \theta_l |x_i^l - x^l|^{p_l}\right) \quad (2.3)$$

gegeben, was der Vermutung entspringt, daß Zielfunktionswerte weit auseinanderliegender Punkte weniger stark korrelieren, während sie bei geringerer Entfernung stärker in Verbindung stehen. Die Bezeichnung ϕ wurde dabei mit Absicht gewählt, weil sich herausstellen wird, daß dieser Ansatz zur Interpolation mit

Radialen Basisfunktionen führt.
Der Zufallsvektor $(Z(x_1), \ldots, Z(x_m))^T$ mit dem Erwartungswert $\vec{1}\mu$ und der symmetrisch positiv definiten Kovarianzmatrix $\sigma^2 R$ modelliert also die Zielfunktion in den Auswertungsstellen $x_1, \ldots, x_m$. Dabei bezeichnet $\vec{1}$ den Vektor $(1, \ldots, 1)^T$ mit zu Y passender Dimensionierung.. Dieses Modell hängt ab von den Parametern $\mu, \sigma, \theta_l, \ldots \theta_n, p_1, \ldots, p_n$. Wir wollen nun herausfinden, für welche Wahl dieser Parameter die in Ξ gegebenen Y am plausibelsten sind.
Dazu benötigen wir die Likelihoodfunktion

$$\frac{1}{(2\pi)^{\frac{n}{2}}(\sigma^2)^{\frac{n}{2}}|R|^{\frac{1}{2}}} \exp\left(\frac{-(Y-\vec{1}\mu)^T R^{-1}(Y-\vec{1}\mu)}{2\sigma^2}\right), \tag{2.4}$$

$|R|$ ist die Determinante von R. Die Maximierung von (2.4) bezüglich μ, σ, θ_l und p_l für $l = 1, \ldots, n$ liefert die Parameter $\hat{\mu}, \hat{\sigma}^2, \hat{\theta}_l, \hat{p}_l$, $l = 1, \ldots, n$, und damit die Korrelationsmatrix $\hat{R}$, die wie in (2.3), nur mit den Parametern $\hat{\theta}_l, \hat{p}_l, l = 1, \ldots, n$, gebildet wird. Dem durch diese Parameter beschriebenen Funktionsmodell läßt sich das Auftreten der beobachteten Auswertungen besser zuordnen als jedem durch andere Parameter definierten Modell.
Es seien nun konkrete Wert für Ξ und Y gegeben. Mit dem Ziel, sinnvolle Aussagen über den Zielfunktionswert an der Stelle $x \notin \Xi$, die wir für diese Betrachtung festhalten, zu machen, berechnen wir zunächst $\hat{\mu}, \hat{\sigma}^2, \hat{\theta}_l, \hat{p}_l$ durch Maximierung von (2.4). Damit ist auch $\hat{R}$ gegeben. Dann raten wir für $f(x)$ zunächst irgendeinen Wert y. Wir erweitern Ξ um x und Y um y. Ensprechend (2.3) erhalten wir eine erweiterte Korrelationsmatrix $\tilde{R}$, die definiert ist durch

$$\tilde{R} = \begin{pmatrix} \hat{R} & r \\ r^T & 1 \end{pmatrix} \quad \text{mit} \quad r = \begin{pmatrix} \phi_1(x) \\ \vdots \\ \phi_m(x) \end{pmatrix}, \tag{2.5}$$

womit wir analog zu (2.4) die Likelihoodfunktion

$$\frac{1}{(2\pi)^{\frac{n}{2}}(\hat{\sigma}^2)^{\frac{n}{2}}|\hat{R}|^{\frac{1}{2}}} \exp\left(\frac{-(Y-\vec{1}\hat{\mu})^T \hat{R}^{-1}(Y-\vec{1}\hat{\mu})}{2\hat{\sigma}^2}\right) \tag{2.6}$$

aufstellen, die Auskunft darüber gibt, wie gut wir y geraten haben. Der plausibelste Wert für y ist demnach der, der (2.6) maximiert. Führt man dies durch, so erhält man als Voraussage $\Phi(x)$ für den Zielfunktionswert an der Stelle x den Wert

$$\Phi(x) := \hat{\mu} + r^T R^{-1}(Y - 1\hat{\mu}) = a + \sum_{i=1}^{m} \lambda_i \phi_i(x). \tag{2.7}$$

a ist eine Konstante, die λ_i sind aus $\mathbb{R}$ und die ϕ_i wie in (2.3), es gehen die Parameter $\hat{\theta}_l, \hat{p}_l$, $l = 1, \ldots, n$, ein. Damit ist die Verbindung zu den Radialen Basisfunktionen hergestellt.
Mit der erwähnten Statistik haben wir nun die Möglichkeit, eine Formel für den mittleren quadratischen Fehler von Φ an der Stelle x anzugeben:

$$s^2(x) = \hat{\sigma}^2 \left(1 - r^T \hat{R}^{-1} r + \frac{\left(1 - r^T \hat{R}^{-1} r\right)^2}{\vec{1}^T \hat{R}^{-1} \vec{1}} \right). \tag{2.8}$$

Dabei hat s^2 die erwartete Eigenschaft, auf Ξ zu verschwinden. Hierauf lassen sich verschiedene Verfahrensvorschriften aufbauen. Die bekanntesten seien hier vorgestellt.

2.3.1 Minimierung einer statistischen Untergrenze

Durch (2.8) hat man ein Instrument, mit dem eine „Statistische Untergrenze“ erzeugt werden kann. Sie beschreibt, mit welcher Wahrscheinlichkeit sich der Zielfunktionswert zum jeweiligen Punkt in einem gewissen Bereich befindet. Deswegen ist es naheliegend, ein globales Minimum dieser Unterschranke

$$\Phi_u(x) := \Phi(x) - \kappa s^2(x) \tag{2.9}$$

zu betrachten, was zur folgenden Iteration führt:

Algorithmus 3 Minimierung Statistischer Untergrenze
Input: Ξ, Y, f **Output:** x

1: Definiere $\Xi_0 := \Xi$ und $Y_0 := Y$
2: i:=0
3: **while** Abbruchkriterium nicht erfüllt **do**
4: Bestimme Interpolanden Φ zu Ξ_i und Y_i nach (2.3) - (2.7)
5: Bestimme mittleren Quadratischen Fehler s^2 nach (2.8)
6: Wähle κ und bestimme Unterschranke $\Phi_u = \Phi - \kappa s^2$ wie in (2.9)
7: Finde globales Minumum x von Φ_u
8: Setze $\Xi_{i+1} = \Xi_i \cup x$ und $Y_{i+1} = Y_i \cup f(x)$
9: i:=i+1
10: **end while**

Wie man in Abbildung 2.3, wo die Schritte eins bis vier eben beschriebener Methode durchgeführt wurden, sieht, ist dieses Verfahren in der Lage, einem nichtglobalen Minimum zu entweichen. Es wird dennoch für ungeeignetes κ keine globale Konvergenz erzielt [27]. Dies ist aber nicht der Hauptgrund, aus dem auf diesen Ansatz nicht zurückgegriffen wird. Mehr dazu im Abschnitt über einstufige Methoden.
Es sei nur noch erwähnt, daß der DIRECT Algorithmus aus [25] gewissermaßen in dieses Schema paßt: Dort wird durch Annahme verschiedener Lipschitzkonstanten die Zielfunktion nach unten abgeschätzt. Obwohl die Methode nicht für extrem teure Zielfunktionen entwickelt wurde, benötigte sie relativ wenige Auswertungen. Deshalb wird sie auch heute noch zum Vergleich herangezogen.
Nun seien noch weitere Möglichkeiten untersucht, die durch die Betrachtung der Zielfunktion als statistisches Modell eröffnet werden.

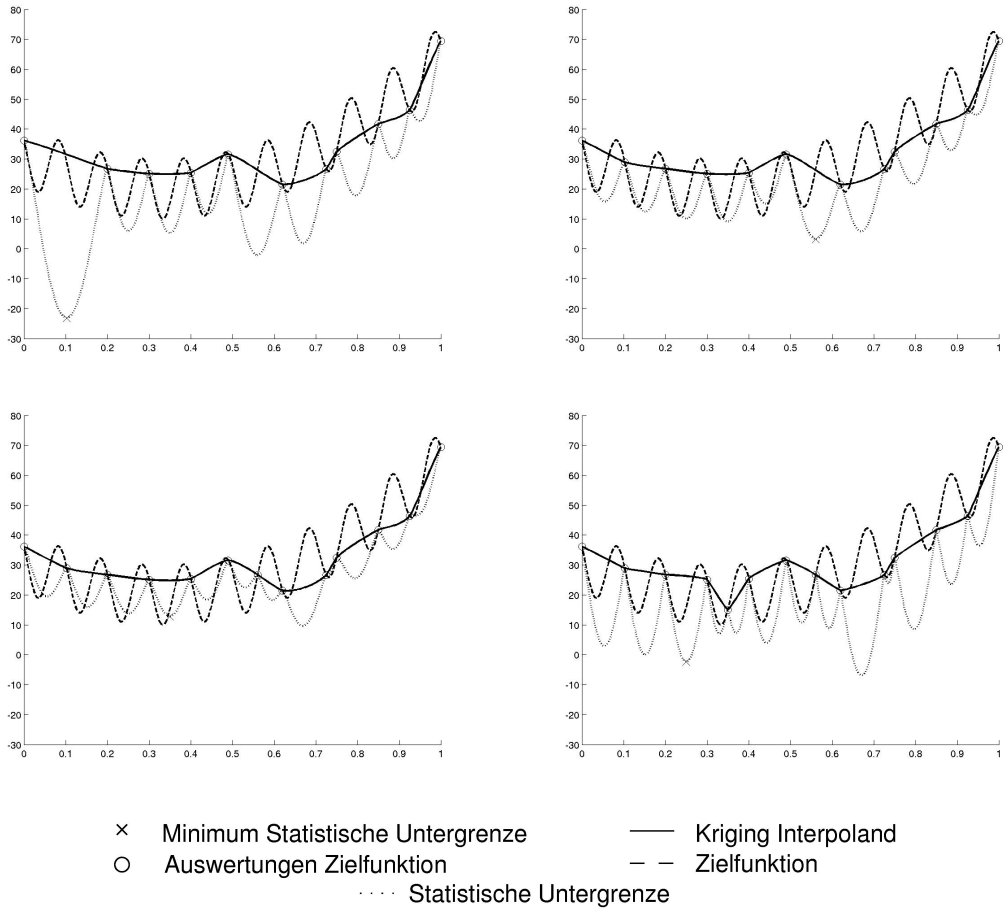


Abbildung 2.3: Minimierung Statistische Untergrenze

2.3.2 Maximierung der Verbesserungswahrscheinlichkeit

Anstatt die Funktion nach unten zu begrenzen, untersuchen wir nun, mit welcher Wahrscheinlichkeit die Zielfunktion in x eine Untergrenze U unterbietet. Nach (2.7) erwarten wir dort den Wert $\Phi(x)$ und gehen von der Standardabweichung $s(x)$ entsprechend (2.8) aus. Da wir eine Normalverteilung annehmen, ist die Wahrscheilichkeit $P_U(x)$, in x den Wert U zu unterbieten, gegeben durch

$$P_U(x) = \Phi_N \left(\frac{U - \Phi(x)}{s(x)} \right), \tag{2.10}$$

wobei Φ_N die Verteilungsfunktion der Normalverteilung ist. Die Schranke U wird nun am wahrscheinlichsten im globalen Maximum x^* von $P_U(x)$ unterboten. Das Ergebnis dieser Überlegung ist

Algorithmus 4 Maximierung Verbesserungswahscheinlichkeit
Input: Ξ, Y, f **Output:** x

1: Definiere $\Xi_0 := \Xi$ und $Y_0 := Y$
2: i:=0
3: **while** Abbruchkriterium nicht erfüllt **do**
4: Bestimme Interpolanden Φ zu Ξ_i und Y_i nach (2.3) - (2.7)
5: Bestimme Standardabweichung s nach (2.8)
6: Bestimme geeignete Untergrenze U
7: Bestimme P_U nach (2.10)
8: Finde globales Maximum x von P_U
9: Setze $\Xi_{i+1} = \Xi_i \cup x$ und $Y_{i+1} = Y_i \cup f(x)$
10: i:=i+1
11: **end while**

Unter geringfügigen Voraussetzungen ist Algorithmus 4 global konvergent, was in [23] für eine etwas allgemeiner fomulierte Methode nachgelesen werden kann. Abbildung 2.4 zeigt Algorithmus 4 bei der Durchführung der Schritte eins bis drei und fünf. Die Untergrenze wurde in jedem Schritt nach der Fomel $U := y_{min} - \frac{1}{4}|y_{min}|$ bestimmt, wobei y_{min} den bisher kleinsten Zielfunktionswert auf Ξ bezeichnet. Das Verhalten von Algorithmus 4 hängt unmittelbar davon ab, wie U gewählt wird. Verlangt man eine vergleichsweise geringfügige Verbesserung, tendiert das Verhalten dahin, lange in der unmittelbaren Nähe des bisher besten Punktes zu suchen, bevor in Bereichen ausgewertet wird, in denen die Auswertungen weniger konzentriert vorliegen. Werden dagegen erhebliche Verbesserungen gefordert, neigt das Verfahren zur Auffüllung des gesamten Suchbereichs, die Ziel-

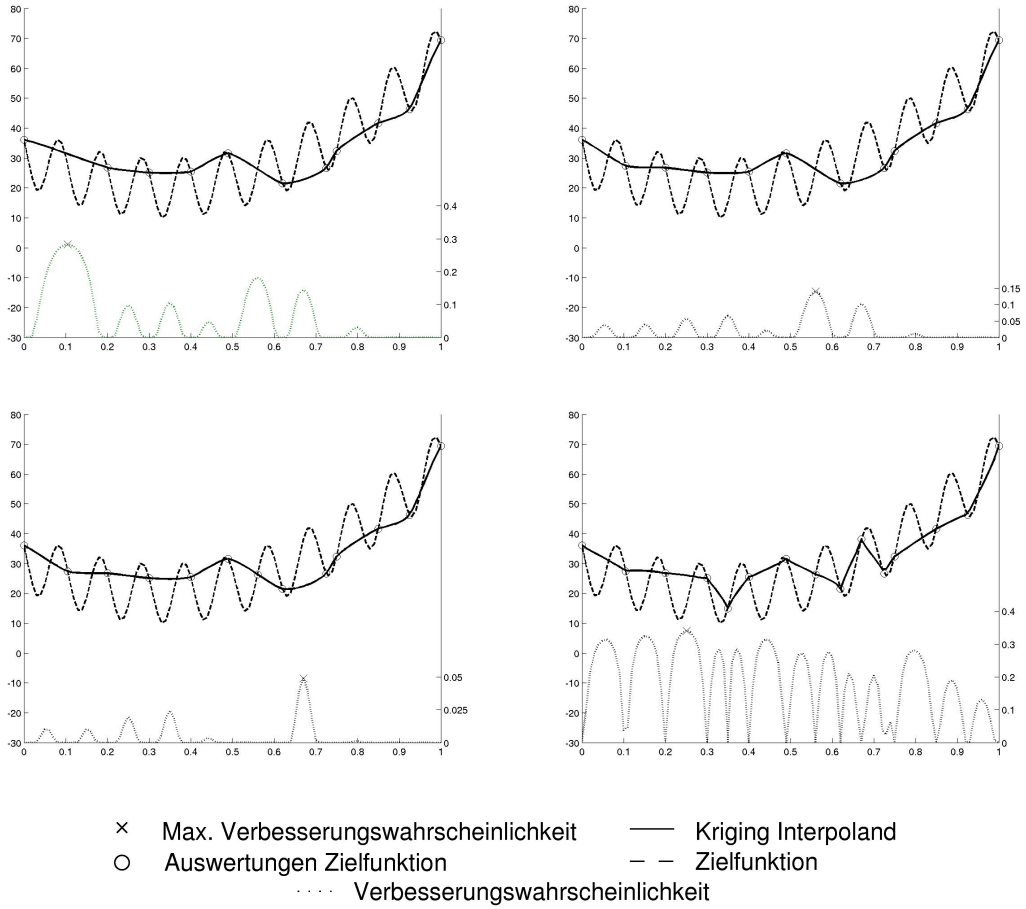


Abbildung 2.4: Maximierung Verbesserungswahrscheinlichkeit

funktionswerte spielen eine untergeordnete Rolle. Verfahren, die nach diesem Prinzip konstruiert sind, finden in der Forschung derzeit keine Beachtung.

2.3.3 Maximierung der zu erwartenden Verbesserung

Den Abschnitt über Kriging-Methoden schließen wir ab mit der Überlegung, die in x zu erwartende Verbesserung zu maximieren. Dafür sehen wir wie im vorangegangenen Abschnitt den Zielfunktionswert in x als normalverteilt an mit Erwartungswert $\Phi(x)$ nach (2.7) und Varianz $s^2(x)$ nach (2.8). Den bisher besten Funktionswert y_{min} wollen wir um V verbessern, was zu der Dichtefunktion

$$\frac{1}{\sqrt{2\pi}s(x)} \exp\left(-\frac{(y_{min} - V - \Phi(x))^2}{2s(x)^2}\right)$$

führt. Daraus erhält man als zu erwartende Verbesserung

$$E(x) = \int_0^\infty V \left(\frac{1}{\sqrt{2\pi}s(x)} \exp\left(-\frac{(y_{min} - V - \Phi(x))^2}{2s(x)^2} \right) \right) dV.$$

Mit partieller Integration leitet man

$$E(x) = s(x)\left(u\Phi_N(u) + D_N(u)\right) \tag{2.11}$$

her, wobei

$$u = \frac{y_{min} - \Phi(x)}{s(x)},$$

und mit Φ_N und D_N jeweils die Verteilungs- bzw. Dichtefunktion der Normalverteilung gemeint ist.
Natürlich betrachtet man nun den Punkt als den interessantesten, wo E den größten Wert annimmt. Dort wird dann auch ausgewertet, was sich niederschlägt in

Algorithmus 5 Maximierung erwarteter Verbesserung
Input: Ξ, Y,f **Output:** x

1: Definiere $\Xi_0 := \Xi$ und $Y_0 := Y$
2: i:=0
3: **while** Abbruchkriterium nicht erfüllt **do**
4: Bestimme Interpolanden Φ zu Ξ_i und Y_i nach (2.3) - (2.7)
5: Bestimme Standardabweichung s nach (2.8)
6: Bestimme y_{min}
7: Bestimme E nach (2.11)
8: Finde globales Maximum x von E
9: Setze $\Xi_{i+1} = \Xi_i \cup x$ und $Y_{i+1} = Y_i \cup f(x)$
10: i:=i+1
11: **end while**

In Abbildung 2.5 wird die Situation nach Durchführung der ersten vier Schritte von Algorithmus 5 gezeigt. Man sieht, daß auch dieses Verfahren aus lokalen Minima heraus finden kann.
Unter bestimmten Voraussetzungen findet sogar globale Konvergenz statt [29].

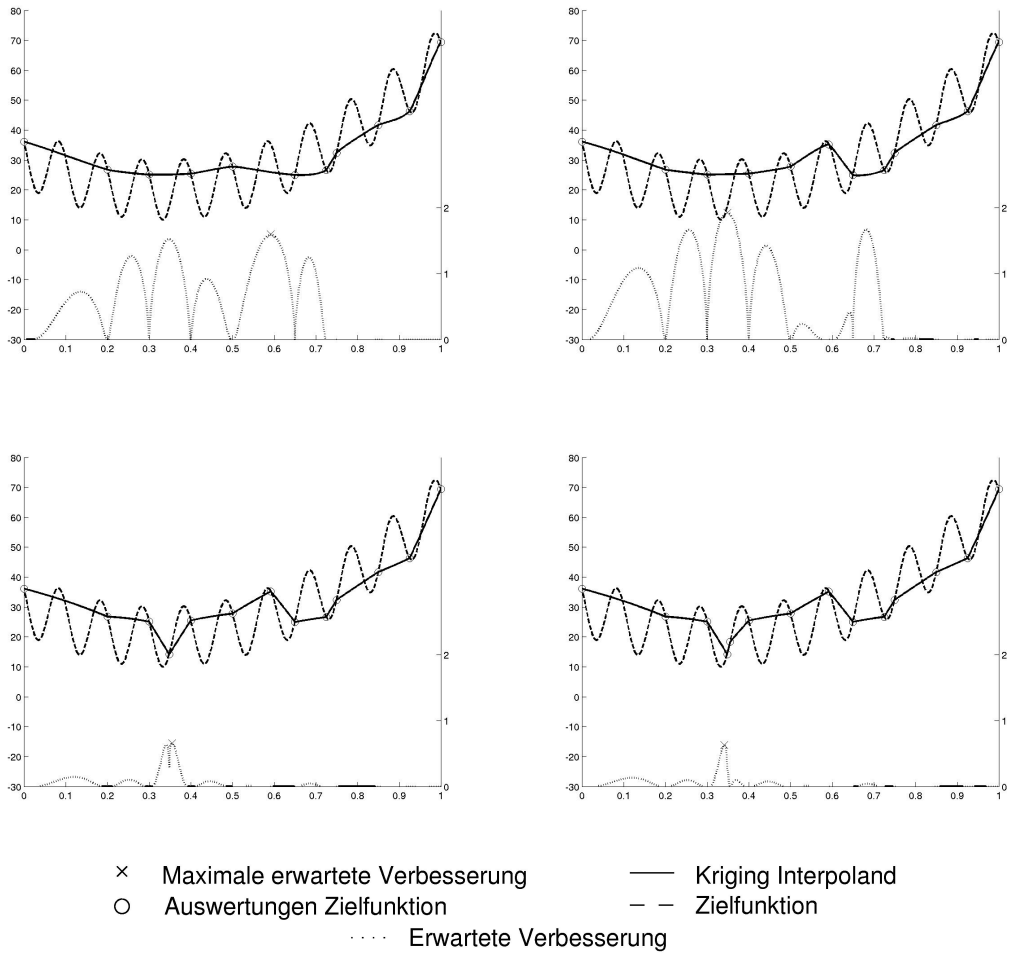


Abbildung 2.5: Maximierung erwarteter Verbesserung

Der EGO-Algorithmus [26], der heute noch und damit auch in dieser Arbeit als Vergleich herangezogen wird, funktioniert nach diesem Prinzip. Die globale Konvergenz wird in angegebener Quelle vermutet.
Allerdings hat dieses Verfahren Schwierigkeiten, wenn der Funktionsverlauf das globale Minimum nicht so gut andeutet wie im gewählten Beispiel. Die Standardabweichung wird dann zu klein geschätzt und erst nach langer Suche wird auch in scheinbar weniger aussichtsreichen Gebiete ausgewertet.
Alle bisher beschriebenen Algorithmen haben gemeinsam, daß sie in einer ersten Stufe aufgrund beobachteter Auswertungen versuchen, die Funktion möglichst gut wiederzugeben. Darauf basierend wird in der zeiten Stufe nach einem möglichst interessanten Punkt für die nächste Auswertung gesucht.
Eine Schwachstelle der Algorithmen 3, 4 und 5 ist also, daß sie sich darauf verlassen, daß der gegebene Interpoland die Funktion gut wiedergibt. Abhilfe kann

man dadurch schaffen, daß man die Parameter, die in Φ eingehen, nicht nur von den beobachteten Daten abhängig macht. Vielmehr soll mit den aus der Statistik erwähnten Mitteln untersucht werden, wie gut das Erreichen eines globalen Minimums an einer bestimmten Stelle zu den beobachteten Werten paßt. Dies machen sogenannte einstufige Methoden.

2.4 Einstufige Methoden

Im Gegensatz zu den eben beschriebenen zweistufigen Methoden beziehen wir nun den Knoten x^z, der auf seine Interessantheit untersucht werden soll, in die Modellierung der Funktion ein. Unter der Bedingung, daß in x^z der Wert y^z erreicht wird, erhalten wir die Likelihoodfunktion

$$L(x^z, y^z, \mu, \sigma, \theta_l, \dots \theta_n, p_1, \dots, p_n) = \frac{\exp\left(\frac{-(Y-\tilde{\mu})^T C^{-1}(Y-\tilde{\mu})}{2\sigma^2}\right)}{(2\pi)^{\frac{n}{2}}(\sigma^2)^{\frac{n}{2}}|C|^{\frac{1}{2}}}, \tag{2.12}$$

wobei

$$\tilde{\mu} = \vec{1}\mu + r(y^z - \mu) \quad \text{und} \quad C = R - rr^T \tag{2.13}$$

mit

$$r = \begin{pmatrix} \phi_1(x^z) \\ \vdots \\ \phi_m(x^z) \end{pmatrix}, \tag{2.14}$$

Sie drückt also aus, wie gut die Daten Ξ, Y zu dem durch $\mu, \sigma, \theta_l, \dots \theta_n, p_1, \dots, p_n$ beschriebenen Modell passen unter der Voraussetzung, daß auch x^z, y^z interpoliert wird. (2.12) ist eine Funktion, die von x^z und y^z abhängt, x^z geht durch r ein, y^z durch $\tilde{\mu}$. Fixieren wir x^z in (2.12) und maximieren L bezüglich $\mu, \sigma, \theta_l, \dots \theta_n, p_1, \dots$ $\dots, p_n$, so erhalten wir ein Maß für die Interessantheit dieses Punktes:

$$I_{y^z}(x^z) := \underset{\mu, \sigma, \theta_l, \dots \theta_n, p_1, \dots, p_n}{\operatorname{argmax}} L(x^z, y^z, \mu, \sigma, \theta_l, \dots \theta_n, p_1, \dots, p_n) \tag{2.15}$$

Am interessantesten ist dann natürlich die Stelle, die I_{y^z} maximiert. Darauf baut auf:

Algorithmus 6 Einstufiges Verfahren

Input: Ξ, Y,f **Output:** x

1: Definiere $\Xi_0 := \Xi$ und $Y_0 := Y$
2: i:=0
3: **while** Abbruchkriterium nicht erfüllt **do**
4: Finde geeignetes y^z
5: Bestimme I_{y^z} nach (2.12) bis (2.15)
6: $x = \text{argmax}_{x \in \mathcal{D}} I_{y^z}(x)$
7: Setze $\Xi_{i+1} = \Xi_i \cup x$ und $Y_{i+1} = Y_i \cup f(x)$
8: i:=i+1
9: **end while**

Wir gehen von der Anfangsverteilung und Zielfunktion aus, die wir auch zum Test der Algorithmen 1 bis 5 verwendet haben. Die ersten vier Iterationen von Algorithmus 6 zeigt Abbildung 2.6.
Die Wahl eines geeigneten y^z ist leider nicht einfach. Kennt man den global minimalen Wert nicht, was die Regel ist, kann man sich zum Beispiel dadurch helfen, daß man eine Reihe von verschiedenen y^z vorgibt und für diese die optimalen $\theta_l, \ldots \theta_n, p_1, \ldots, p_n$ findet. In der Praxis beobachtet man oft eine Häufung der optimalen Parameter trotz verschiedener y^z, siehe [27] für ein Rechenbeispiel.
Ein sehr gutes einstufiges Verfahren wird in verschiedenen Implementierungen als rbfSolve in [3] und RBF-G in [23] vorgestellt. Wie gut ein neues Paar x^z, y^z zur bisherigen Menge von Knoten und Auswertungen Ξ_i, Y_i paßt, wird dort durch die Funktion

$$g(x^z) := \mu(x^z)(\Phi(x^z) - y^z)^2$$

ausgedrückt. Dies ist so zu verstehen: Sei Φ der Interpoland zu Ξ_i, Y_i und Φ^z der zu $\{\Xi_i, x^z\}, \{Y_i, y^z\}$. Φ habe die Form

$$\Phi(x) = \sum_{i=1}^{m} \lambda_i \phi_i(x) + b^T x + a.$$

Dann wird $\sigma(\Phi)$ definiert als

$$\sigma(\Phi) := \sum_{i=1}^{m} \lambda_i \Phi(x_i).$$

Die Interessantheit des Paares x^z, y^r ist dann definiert als $\sigma(\Phi^z)$. Die Minimierung von $\sigma(\Phi^z)$ über alle x^z ist gleichwertig zur Minimierung von $g(x^z)$. Dabei geht

$\mu(x^z)$ hervor aus dem Interpolanden Φ_μ zum Paar $\{\Xi_i, x^z\}, \{0, \ldots, 0, 1\}$, der die Form

$$\Phi_\mu = \sum_{i=1}^{m} \lambda_i^\mu \phi(x - x_i) + \mu(x^z)\phi(x - x^z) + b_\mu^T x + a_\mu.$$

hat. Dieser Ansatz macht nur Sinn für

$$y^z \in \left(-\infty, \min_{x \in \mathcal{D}} \Phi(x)\right).$$

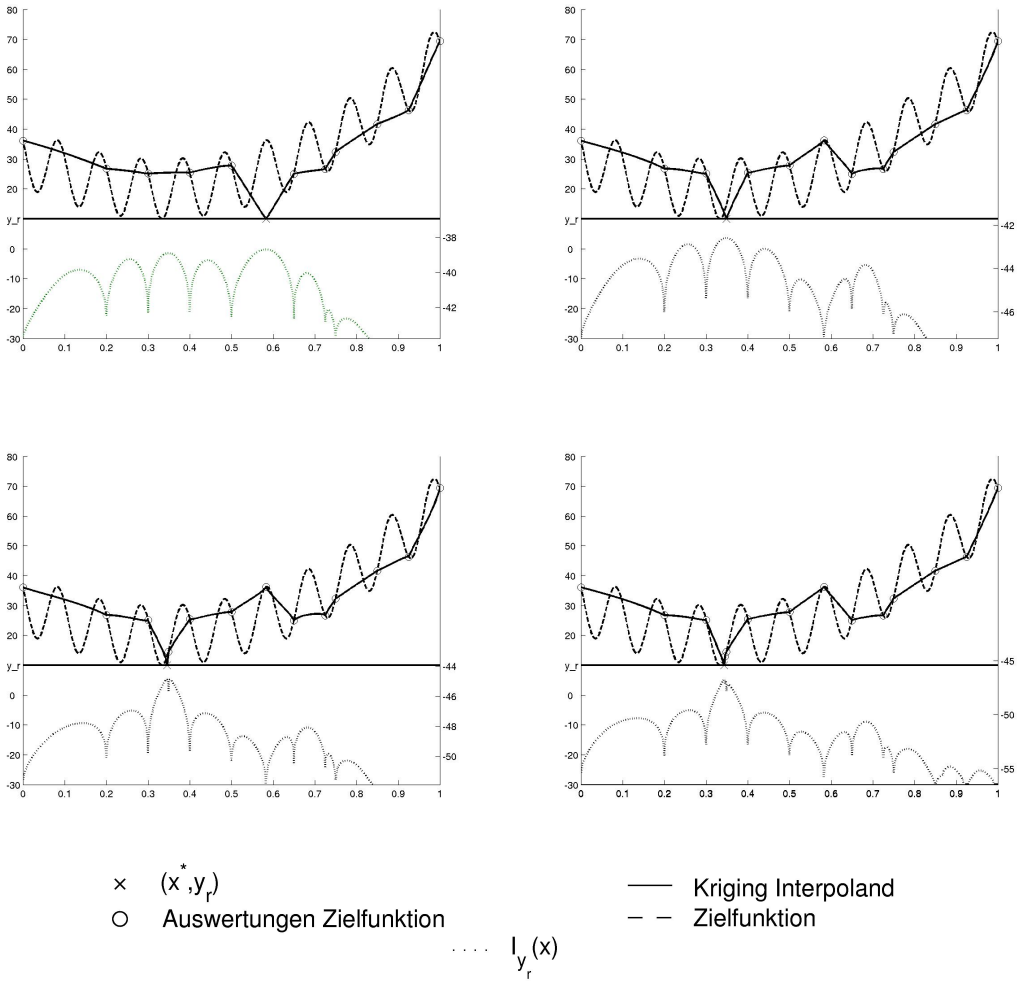


Abbildung 2.6: Einstufiges Verfahren

$y^z = \min_{x \in \mathcal{D}\Phi(x)}$ erzeugt eine reine lokale Suche, je mehr man y^z gegen $-\infty$ gehen läßt, desto weniger gehen die Zielfunktionswerte ein und es wird nur noch

in Bereichen gesucht, in denen bisher kaum ausgewertet wurde. Deshalb wird ähnlich wie im CORS Verfahren im Laufe der Iteration immer wieder ein Zyklus durchlaufen, um beide Formen der Suche zu berücksichtigen. Unter gewissen Voraussetzungen ist dieses Verfahren global konvergent.
Trotzdem bleibt ein wesentlicher Nachteil aller hier aufgeführten Methoden, die auf dem Kriging aufbauen. Besonders deutlich wird er bei der Verwendung einstufiger Verfahren: In unserer Anwendung sollen in jeder Iteration mehrere, sogar einige hundert, neue Knoten gefunden werden. Es ist klar, daß ein vernünftiges Verfahren die bisher festgelegten Knoten und die dort vermuteten Funktionswerte in irgendeiner Form berücksichtigen muß. Also sind in unserem Fall mehrere tausend Kriging Interpolanden zu berechnen. Dies erfordert die Lösung von Linearen Gleichungssystemen der Form (1.9). Leider verändert die Hinzunahme eines neuen Knoten die optimalen $\theta_l, \ldots \theta_n, p_1, \ldots, p_n$, und damit in aller Regel auch sämtliche Einträge der LGS-Matrix, also muß man die vollbesetzten Gleichungssysteme immer wieder von vorne lösen, ohne auf alte Lösungen zurückgreifen zu können. Deshalb sind die auf dem Kriging basierenden Verfahren in ihrer vorgestellten Form für uns nur nach enstprechender Veränderung brauchbar.
Dieser Überblick hat nicht den Anspruch der Vollständigkeit. Er kann nur eine grobe Klassifikation und Übersicht über das liefern, was bisher in der Forschung erarbeitet wurde.

Kapitel 3

Lokale Suchverfahren

Globale Optimierung kann auch lokale Suche beinhalten. Wohl am beliebtesten in dieser Hinsicht ist das Newton-Verfahren. Man schätzt besonders die Eigenschaft, daß man in vergleichsweise wenigen Schritten sehr nahe an das lokale Minimum kommt, was wir gleich noch präziser erläutern. Allerdings benötigt man in jedem Schritt die Hessematrix der Zielfunktion im aktuellen Knoten. Diese liegt meist nicht vor. Man kann sich dann z.B. durch eine Approximation mittels Differenzenquotienten helfen. Dies treibt aber den Aufwand sofort in die Höhe. Daher wurden etliche Abwandlungen entwickelt, die nur Gradientenauswertungen benötigen [36]. All diese Verfahren haben gemeinsam, daß man versucht, den rechnerischen Aufwand in jedem Schritt so gering wie möglich zu halten. Wir aber können in unserem Fall ohne Bedenken viel Rechenzeit investieren, wenn nur die Anzahl der Iterationen abnimmt. Deshalb machen wir es uns zum Ziel nach Verfahren zu suchen, die möglichst schneller als das Newton-Verfahren ein lokales Minimum finden und im Rahmen der gegebenen Problemstellung durchführbar sind.

3.1 Verallgemeinerung des Newton-Verfahrens

Will man ein Verfahren verbessern bzw. es übertreffen, ist es sicher hilfreich zu verstehen, wie es funktioniert. In unserem Fall wird sich aus der Herleitung des Newton-Verfahrens in seiner einfachsten Fassung unmittelbar eine mögliche Verallgemeinerung ergeben. Wir nehmen an dieser Stelle vorweg, daß man zwei mal auf beschriebene Weise vorgehen kann, bevor man an die Grenzen der Problemstellung stößt. Mehr dazu in Abschnitt 3.2.
Unter geeigneten Differenzierbarkeitsvoraussetzungen für die Zielfunktion f gilt

$$f(x^k + h) = f(x^k) + \nabla f(x^k)^T h + \frac{1}{2} h^T H_f(x^k) h + \mathcal{O}(\|h\|^3), \tag{3.1}$$

und wir erhalten damit die quadratische Approximation

$$T_2(x^k + h) := f(x^k) + \nabla f(x^k)^T h + \frac{1}{2} h^T H_f(x^k) h,$$

was Abbildung 3.1 an einem Beispiel verdeutlicht.

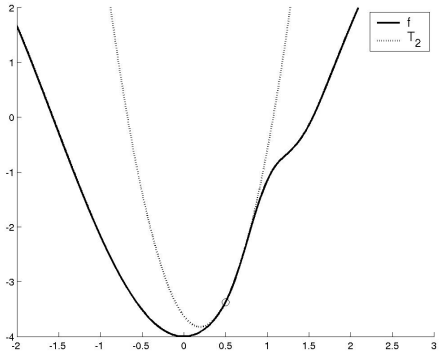


Abbildung 3.1: Quadratische Approximation der Zielfunktion

Statt des Problems

$$\min_h f(x^k + h)$$

lösen wir nun

$$h_k^* := \text{argmin}_h T_2(x^k + h).$$

Notwendig dafür ist das Auflösen von $\nabla_h T_2(x^k + h) = 0$. Es ergibt sich

$$0 \stackrel{!}{=} \nabla_h T_2(x^k + h) = \nabla f(x^k) + H_f(x^k) h.$$

Umstellen liefert die Suchrichtung des Newton-Verfahrens:

$$\boxed{h_k^* = -H_f^{-1}(x^k) \nabla f(x^k).} \tag{3.2}$$

Für positiv definites $H_f(x^k)$ ist $T_2(x^k + h_k^*)$ minimal. Die beschriebene Iteration wird in $x^{k+1} := x^k + h_k^*$ wie zuvor fortgesetzt. Es gilt dann der

Satz 3.1. *Sei f dreimal stetig differenzierbar, $x^* \in \mathbb{R}^n$ mit $\nabla f(x^*) = 0$ und $H_f(x^*)$ in x^* invertierbar. Dann ist das Newton-Vefahren für Startwerte nahe genug an x^* durchführbar und konvergiert gegen x^* mit der Eigenschaft*

$$\|x^{k+1} - x^*\| = \mathcal{O}\left(\|x^k - x^*\|^2\right)$$

Beweis: Siehe [6]. □

Man spricht hier von quadratischer Konvergenz, was bedeutet, daß sich im wesentlichen die Zahl der gültigen Dezimalstellen mit jeder Iteration verdoppelt, wenn man einmal nahe genug an der Lösung ist. Dies erklärt, warum das Newton-Verfahren unter Anwendern so beliebt ist.
Verallgemeinern läßt sich dieses Vorgehen z.B. dadurch, daß man analog für T_3, T_4 usw. vorgeht. Im Folgenden wird erklärt, wie dies geschehen kann und welche Schwierigkeiten auftreten können.
Die naheliegendste Verallgemeinerung ist, die Zielfunktion nicht nur quadratisch, sondern sogar kubisch zu approximieren. Wir bilden also

$$T_3(x^k + h) := f(x^k) + \nabla f(x^k)^T h + \frac{1}{2} h^T H_f(x^k) h + \frac{1}{6} \sum_{i,j,l} \frac{\partial^3 f(x^k)}{\partial x_i x_j x_l} h_i h_j h_l, \quad (3.3)$$

was im Eindimensionalen etwa wie in Abbilung 3.2 aussieht.

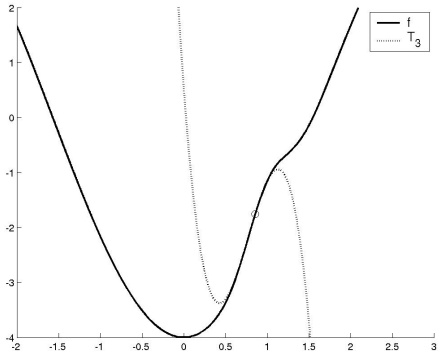


Abbildung 3.2: Kubische Approximation der Zielfunktion

Man erkennt sofort, daß in der Regel die Gradientennullstelle der Approximation nicht eindeutig sein wird. Trotzdem definieren wir für $l = 1, \ldots, n$

$$T_f^l(x^k, h) := \frac{1}{2} \sum_{i,j} \frac{\partial^3 f(x^k)}{\partial x_l x_i x_j} h_i h_j$$

und daraus

$$T_f(x^k, h) := \left(T_f^1(x^k, h), \ldots, T_f^n(x^k, h)\right)^T .$$

Damit errechnet sich

$$\nabla_h T_3(x^k + h) = \nabla f(x^k) + H_f(x^k)h + T_f(x^k, h).$$

Wie bereits in Abbildung 3.2 angedeutet, läßt sich $\nabla_h T_3(x^k + h) = 0$ in der Regel nicht ohne weiteres nach h auflösen. Es ist aber möglich, $T_f(x^k, h)$ durch $T_f(x^k, h_k^*)$ zu ersetzen, wobei h_k^* wie in (3.2) berechnet wurde, und dann

$$0 = \nabla f(x^k) + H_f(x^k)h + T_f(x^k, h_k^*).$$

nach h umzustellen. Man erhält die Suchrichtung des sogenannten Halley-Verfahrens [10]

$$\boxed{\tilde{h}_k = -H_f^{-1}(x^k)\left(\nabla f(x^k) + T_f(x^k, h_k^*)\right).} \tag{3.4}$$

Wir werden später noch zeigen, daß man mit der Iterationsvorschrift

$$x^{k+1} := x^k + \tilde{h}_k \tag{3.5}$$

kubische Konvergenz erzielt. Dadurch ermutigt versucht man ein ähnliches Vorgehen für

$$T_4(x^k + h) = f(x^k) + \nabla f(x^k)^T h + \frac{1}{2}h^T H_f(x^k)h + \frac{1}{6}\sum_{i,j,l}\frac{\partial^3 f(x^k)}{\partial x_i x_j x_l}h_i h_j h_l$$

$$+\frac{1}{24}\sum_{i,j,l,t}\frac{\partial^4 f(x^k)}{\partial x_i x_j x_l x_t}h_i h_j h_l h_t,$$

was wir beispielhaft in Abbildung 3.3 sehen. Bezüglich der Minimierung von T_4 deuten sich ähnliche Probleme wie für T_3 an.

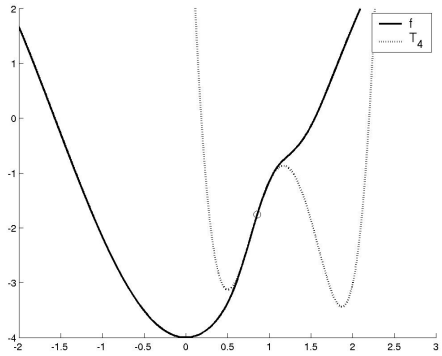


Abbildung 3.3: Approximation der Zielfunktion durch Polynom vierten Grades

Wir definieren für $l = 1, \ldots, m$

$$Q_f^l(x^k, h) := \frac{1}{6} \sum_{i,j,t} \frac{\partial^4 f(x^k)}{\partial x_l x_i x_j x_t} h_i h_j h_t,$$

und darauf aufbauend

$$Q_f(x^k, h) := \left(Q_f^1(x^k, h), \ldots, Q_f^n(x^k, h)\right)^T.$$

Man errechnet

$$\nabla_h T_4(x^k + h) = \nabla f(x^k) + H_f(x^k)h + T_f(x^k, h) + Q_f(x^k, h).$$

Es ist erneut nicht offensichtlich, wie man eine Gleichung des Typs

$$\nabla_h T_4(x^k + h) = 0$$

nach h auflösen kann, deswegen verfährt man ähnlich wie oben, und stellt

$$0 = \nabla f(x^k) + H_f(x^k)h + T_f(x^k, \tilde{h}_k) + Q_f(x^k, \tilde{h}_k).$$

mit $\tilde{h}_k$ wie in (3.4) nach h um. Das liefert

$$\boxed{\hat{h}_k = -H_f^{-1}(x^k)\left(\nabla f(x^k) + T_f(x^k, \tilde{h}_k) + Q_f(x^k, \tilde{h}_k)\right).} \tag{3.6}$$

Mit der Iterationsvorschrift

$$x^{k+1} = x^k + \hat{h}_k \tag{3.7}$$

erhalten wir das Verfahren $\mathcal{V}$ der Konvergenzordnung vier, was nun gezeigt wird. Als Zwischenresultat wird zuerst noch die Ordnung drei des Halley-Verfahrens gezeigt. Ähnlich wie dieser Beweis läuft dann auch der für $\mathcal{V}$ ab. Um dies möglichst übersichtlich zu gestalten, werden zunächst Notationen geklärt und benötigte Resultate aufgeführt. So definieren wir

$$N(x^k) := x^k - H_f^{-1}(x^k)\nabla f(x^k).$$

Liegt x^k nahe genug an der Gradientennullstelle x^* so gilt wegen Satz 3.1

$$\|N(x^k) - x^*\| = \mathcal{O}\left(\|x^k - x^*\|^2\right) \tag{3.8}$$

Mit der Taylorentwicklung von H_f erhalten wir

$$H_f(x^k) = H_f(x^*)+\mathcal{O}(\|x^k-x^*\|) \;\Rightarrow\; \|H_f(x^k)-H_f(x^*)\| = \mathcal{O}\left(\|x^k - x^*\|\right), \tag{3.9}$$

Taylorentwicklung von ∇f um x^* liefert (mit $\nabla f(x^*) = 0$)

$$\nabla f(N(x^k)) = H_f(x^*)(N(x^k) - x^*) + \mathcal{O}(\|x^k - x^*\|^4),$$

und damit

$$H_f(x^*)(N(x^k) - x^*) = \nabla f(N(x^k)) + \mathcal{O}(\|x^k - x^*\|^4). \tag{3.10}$$

Erneut wenden wir Taylor an:

$$\begin{aligned} \nabla f(N(x^k)) &= \nabla f(x^k - H_f^{-1}(x^k)\nabla f(x^k)) = \nabla f(x^k) - H_f(x^k)H_f^{-1}(x^k)\nabla f(x^k) \\ &\quad + T_f(x^k, -H_f^{-1}(x^k)\nabla f(x^k)) + \mathcal{O}(\|H_f^{-1}(x^k)\nabla f(x^k)\|^3) \qquad (3.11) \\ &= T_f(x^k, h_k^*) + \mathcal{O}(\|H_f^{-1}(x^k)\nabla f(x^k)\|^3). \end{aligned}$$

Dies ist nur möglich, wenn f viermal stetig differenzierbar ist. Zuletzt werden wir noch benötigen, daß

$$\begin{aligned} \| - H_f^{-1}(x^k)\nabla f(x^k)\| &= \|N(x^k) - x^k\| = \|N(x^k) - x^* - (x^k - x^*)\| \\ &\leq \|N(x^k) - x^*\| + \|x^k - x^*\| = \mathcal{O}(\|x^k - x^*\|). \end{aligned} \tag{3.12}$$

Damit gilt

Satz 3.2. *Es gelten die Voraussetzungen von Satz 3.1, f sei darüber hinaus viermal stetig differenzierbar. Dann ist das Halley-Verfahren für Startwerte nahe genug an x^* durchführbar und konvergiert gegen x^* mit der Eigenschaft*

$$\|x^{k+1} - x^*\| = \mathcal{O}\left(\|x^k - x^*\|^3\right)$$

Beweis: Es sei $\beta := \max_{x^k \in \mathcal{D}} \|H_f^{-1}(x^k)\|$. β ist nach den Voraussetzungen von Satz 3.1 in einer Umgebung von x^* beschränkt. Damit gilt:

$$\begin{aligned}
\|x^k + \tilde{h}_k - x^*\| &= \|x^k - x^* - H_f^{-1}(x^k)\left(\nabla f(x^k) + T_f(x^k, h_k^*)\right)\| \\
&\leq \|H_f^{-1}(x^k)\| \cdot \|H_f(x^k)x^k - H_f(x^k)x^* - \nabla f(x^k) \\
&\qquad -T_f(x^k, h_k^*)\| \\
&= \beta \cdot \|H_f(x^k)\left(x^k - H_f^{-1}(x^k)\nabla f\right) - H_f(x^k)x^* - \\
&\qquad -T_f(x^k, h_k^*)\| \\
&= \beta \cdot \|H_f(x^k)\left(N(x^k) - x^*\right) - T_f(x^k, h_k^*)\| \\
&= \beta \cdot \|\left(H_f(x^k) - H_f(x^*)\right)\left(N(x^k) - x^*\right) - T_f(x^k, h_k^*) \\
&\qquad +H_f(x^*)\left(N(x^k) - x^*\right)\| \\
&\leq \beta \cdot \|H_f(x^k) - H_f(x^*)\| \cdot \|N(x^k) - x^*\| \\
&\quad +\beta \cdot \|H_f(x^*)\left(N(x^k) - x^*\right) - T_f(x^k, h_k^*)\| \\
&\overset{(3.8),(3.9)}{=} \mathcal{O}(\|x^k - x^*\|^3) + \beta \cdot \|H_f(x^*)\left(N(x^k) - x^*\right) \\
&\qquad -T_f(x^k, h_k^*)\| \\
&\overset{(3.10),(3.11)}{=} \mathcal{O}\left(\|x^k - x^*\|^3\right) + \mathcal{O}\left(\|H_f^{-1}(x^k)\nabla f(x^k)\|^3\right) \\
&\overset{(3.12)}{=} \mathcal{O}\left(\|x^k - x^*\|^3\right)
\end{aligned}$$

□

Ähnlich gehen wir nun für $\mathcal{V}$ vor. Wieder werden die benötigten Resultate zuerst angegeben und dann im Laufe der Rechnung referenziert. Benötigt wird das

Lemma 3.3. *Es gelten die Voraussetzungen von Satz 3.2. Dann gilt die Aussage von dort auch für die Verfahrensvorschrift*

$$x^{k+1} = x^k \underbrace{-H_f^{-1}(x^k)\left(\nabla f(x^k) + T_f(x^k, \tilde{h}_k)\right)}_{h_k^0}$$

Beweis: Sei $p_2 \in \mathbb{P}_n^2$ und $h, \Delta h \in \mathbb{R}$. α und β seien Multiindizes aus dem $\mathbb{Z}^n$, α sei von der Form $(0,\ldots,0,2,0,\ldots,0)$, während β die Struktur $(0,\ldots,0,1,0,\ldots$ $\ldots,0,1,0,\ldots,0)$ habe, die in p_2 auftretenden Koeffizienten bezeichnen wir mit λ, also haben wir

$$p_2(x) = \sum_\alpha \lambda_\alpha x^\alpha + \sum_\beta \lambda_\beta x^\beta.$$

Dann gilt

$$|p_2(h) - p_2(h+\Delta h)| = \left| \sum_\alpha \lambda_\alpha (h^\alpha - (h+\Delta h)^\alpha) + \sum_\beta \lambda_\beta (h^\beta - (h+\Delta h)^\beta) \right|$$

Wir greifen einen Multiindex $\tilde{\alpha}$ heraus, die zugehörige Komponente aus h bezeichnen wir mit $h_{\tilde{\alpha}}$, die aus Δh mit $\Delta h_{\tilde{\alpha}}$. Der entsprechende Vorfaktor in p_2 ist $\lambda_{\tilde{\alpha}}$. Es gilt

$$\begin{aligned} \lambda_{\tilde{\alpha}}(h_{\tilde{\alpha}})^2 - \lambda_{\tilde{\alpha}}(h_{\tilde{\alpha}} + \Delta h_{\tilde{\alpha}})^2 &= \lambda_{\tilde{\alpha}}(-2h_{\tilde{\alpha}}\Delta h_{\tilde{\alpha}} + (\Delta h_{\tilde{\alpha}})^2) \\ &\leq |\lambda_{\tilde{\alpha}}| \cdot (2|h_{\tilde{\alpha}}| \cdot |\Delta h_{\tilde{\alpha}}| + |\Delta h_{\tilde{\alpha}}|^2) \end{aligned}$$

Sind h_m bzw. Δh_m die betragsgrößten Komponenten von h bzw. Δh, so erhalten wir

$$\left| \sum_\alpha \lambda_\alpha (h^\alpha - (h+\Delta h)^\alpha) \right| \leq \left(2|h_m| \cdot |\Delta h_m| + |\Delta h_m|^2\right) \cdot \sum_\alpha |\lambda_\alpha|.$$

Für die β verläuft die Rechnung völlig analog, was zur dieser Abschätzung führt

$$|p_2(h) - p_2(h+\Delta h)| \leq \left(2|h_m| \cdot |\Delta h_m| + |\Delta h_m|^2\right) \cdot \left(\sum_\alpha |\lambda_\alpha| + \sum_\beta |\lambda_\beta| \right) \quad (3.13)$$

Es gelte nun $\Delta h = \mathcal{O}(\|h\|^2)$. Dann gilt auch $\Delta h_m = \mathcal{O}(\|h\|^2)$, und ohnehin $h_m = \mathcal{O}(\|h\|)$. Aus (3.13) schließen wir dann

$$|p_2(h) - p_2(h+\Delta h)| = \mathcal{O}(\|h\|^3).$$

Dies werden wir gleich benutzen, denn es gilt

$$\tilde{h}_k = -H_f^{-1}(x^k)(\nabla f + T_f(x^k, h_k^*)) = h_k^* + \mathcal{O}(\|h_k^*\|^2). \quad (3.14)$$

Da $T_f(x^k, h)$ ein Polynom zweiten Grades in h ist, erhalten wir folgende Abschätzung:

$$\|T_f(x^k, h_k^*) - T_f(x^k, \tilde{h}_k)\| = \mathcal{O}(\|h_k^*\|^3) \overset{(3.9)}{=} \mathcal{O}(\|x^k - x^*\|^3) \quad (3.15)$$

Nun können wir die Behauptung zeigen. Die Rechnung verläuft größtenteils analog zu der im Beweis von Satz 3.2, erst ab der drittletzten Gleichung gibt es Unterschiede.

$$
\begin{aligned}
\|x^k + h_k^0 - x^*\| &= \|x^k - x^* - H_f^{-1}(x^k)(\nabla f(x^k) + T_f(x^k, \tilde{h}_k))\| \\
&\vdots \\
&= \mathcal{O}(\|x^k - x^*\|^3) + \beta \cdot \|H_f(x^*)\left(N(x^k) - x^*\right) \\
&\qquad\qquad -T_f(x^k, \tilde{h}_k)\| \\
&\overset{(3.10),(3.11)}{=} T_f(x^k, h_k^*) - T_f(x^k, \tilde{h}_k) + \mathcal{O}\left(\|x^k - x^*\|^3\right) \\
&\overset{(3.15)}{=} \mathcal{O}\left(\|x^k - x^*\|^3\right)
\end{aligned}
$$

□

Analog zu (3.14) leitet man

$$\left\|h_k^0 - \tilde{h}_k\right\| = \mathcal{O}\left(\|h_k^*\|^3\right)$$

und daraus

$$\|T_f(x^k, \tilde{h}_k) - T_f(x^k, h_k^0)\| = \mathcal{O}\left(\|h_k^*\|^4\right) \tag{3.16}$$

bzw.

$$\|Q_f(x^k, \tilde{h}_k) - Q_f(x^k, h_k^0)\| = \mathcal{O}\left(\|h_k^*\|^4\right) \tag{3.17}$$

her. Wir werden benutzen, daß

$$N^0(x^k) := x^k + h_k^0 \;\Rightarrow\; \|N^0(x^k) - x^*\| = \mathcal{O}\left(\|x^k - x^*\|^3\right). \tag{3.18}$$

Ähnlich zu (3.10) und (3.11) gilt dann

$$H_f(x^*)(N^0(x^k) - x^*) = T_f(x^k, h_k^0) - T_f(x^k, \tilde{h}_k) + Q_f(x^k, h_k^0) + \mathcal{O}\left(\|h_k^*\|^4\right), \tag{3.19}$$

was die fünfmalige stetige Differenzierbarkeit von f voraussetzt. Damit erhalten wir

Satz 3.4. *Es gelten die Voraussetzungen von Satz 3.1, f sei darüber hinaus fünfmal stetig differenzierbar. Dann ist $\mathcal{V}$ für Startwerte nahe genug an x^* durchführbar und konvergiert gegen x^* mit der Eigenschaft*

$$\|x^{k+1} - x^*\| = \mathcal{O}\left(\|x^k - x^*\|^4\right)$$

Die Beweisführung ist ähnlich wie die für das Halley- Verfahren.

$$
\begin{aligned}
\left\|x^k + \hat{h}_k - x^*\right\| &= \Big\|x^k - x^* - H_f^{-1}(x^k)\Big(\nabla f(x^k) + T_f(x^k, \tilde{h}_k) \\
&\qquad + Q_f(x^k, \tilde{h}_k)\Big)\Big\| \\
&\leq \left\|H_f^{-1}(x^k)\right\| \cdot \Big\|H_f(x^k)x^k - H_f(x^k)x^* - \nabla f(x^k) \\
&\qquad - T_f(x^k, \tilde{h}_k) - Q_f(x^k, \tilde{h}_k)\Big\| \\
&= \beta \cdot \Big\|H_f(x^k)\Big(x^k - H_f^{-1}(x^k)\Big(\nabla f(x^k) + T_f(x^k, \tilde{h}_k)\Big)\Big) \\
&\qquad - H_f(x^k)x^* - Q_f(x^k, \tilde{h}_k)\Big\| \\
&= \beta \cdot \Big\|H_f(x^k)\left(N^0(x^k) - x^*\right) - Q_f(x^k, \tilde{h}_k)\Big\| \\
&= \beta \cdot \Big\| \left(H_f(x^k) - H_f(x^*)\right)\left(N^0(x^k) - x^*\right) - Q_f(x^k, \tilde{h}_k) \\
&\qquad + H_f(x^*)\left(N^0(x^k) - x^*\right)\Big\| \\
&\leq \beta \cdot \left\|H_f(x^k) - H_f(x^*)\right\| \cdot \left\|N^0(x^k) - x^*\right\| \\
&\qquad + \beta \cdot \Big\|H_f(x^*)\left(N^0(x^k) - x^*\right) - Q_f(x^k, \tilde{h}_k)\Big\| \\
&\overset{(3.9),(3.18)}{=} \beta \cdot \Big\|H_f(x^*)\left(N^0(x^k) - x^*\right) - Q_f(x^k, \tilde{h}_k)\Big\| \\
&\qquad + \mathcal{O}\left(\|x^k - x^*\|^4\right) \\
&\overset{(3.19)}{=} \beta \cdot \Big\|T_f(x^k, h_k^0) - T_f(x^k, \tilde{h}_k) + Q_f(x^k, h_k^0) \\
&\qquad - Q_f(x^k, \tilde{h}_k)\Big\| + \mathcal{O}\left(\|x^k - x^*\|^4\right) \\
&\overset{(3.16),(3.17)}{=} \mathcal{O}\left(\|x^k - x^*\|^4\right)
\end{aligned}
$$

□

Mit dem Halley-Verfahren und $\mathcal{V}$ sind uns also zwei Methoden gegeben, die unter geeigneten Voraussetzungen zumindest in der Theorie schneller konvergieren als das Newton-Verfahren. In der Literatur wird zwar stets die Konvergenz dritter Ordnung des Halley-Verfahrens erwähnt, allerdings war ein Beweis für den beliebig dimensionalen Fall trotz intensiver Suche nicht auffindbar. Zudem wird er für den Beweis der Ordnung vier des neuen Verfahrens benötigt.
Bevor wir zu einer ausführlichen Testreihe des Halley-Verfahrens und $\mathcal{V}$ kommen, wollen wir noch klären, wie sie im Rahmen unserer Aufgabenstellung durchführbar sind.

3.2 Ableitungsapproximation

Besonders in der Numerik partieller Differentialgleichungen benutzt man immer wieder sogenannte Differenzensterne, um Ableitungen anzunähern. Auch wir machen davon Gebrauch. Wir wollen herausfinden, welche Ableitungen wie gut mit welcher Anzahl an Auswertungen approximiert werden können.
Zunächst macht man sich klar, welche Arten von Ableitungen in den Verfahren (3.5) bzw. (3.7) überhaupt auftreten. Wir beginnen mit dem Halley-Verfahren. Dort ist T_3, entwickelt um eine beliebige Stelle $\bar{x}$, interessant. Um besser zu verstehen, welche partiellen Ableitungen auftreten können, schreiben wir $T_3(\bar{x}+h)$ etwas anders als in (3.3). Weil nur Ableitungen bis einschließlich der Ordnung 3 auftauchen, genügt die Betrachtung der ersten drei Richtungen x, y und z. Mit $h = (h_1, \ldots, h_n)$ erhalten wir

$$\begin{aligned} T_3(\bar{x}+h) \;:=\; & f(\bar{x}) + \frac{\partial f(\bar{x})}{\partial x}h_1 + \frac{\partial f(\bar{x})}{\partial y}h_2 + \frac{\partial f(\bar{x})}{\partial z}h_3 + \ldots \\ & + \frac{1}{2}\frac{\partial^2 f(\bar{x})}{\partial x^2}h_1^2 + \frac{\partial^2 f(\bar{x})}{\partial xy}h_1h_2 + \cdots + \frac{\partial^2 f(\bar{x})}{\partial yz}h_2h_3 + \frac{1}{2}\frac{\partial^2 f(\bar{x})}{\partial z^2}h_3^2 + \ldots \\ & + \frac{1}{6}\frac{\partial^3 f(\bar{x})}{\partial x^3}h_1^3 + \frac{1}{2}\frac{\partial^3 f(\bar{x})}{\partial x^2 y}h_1^2h_2 + \frac{1}{2}\frac{\partial^3 f(\bar{x})}{\partial xy^2}h_1h_2^2 + \ldots \\ & + \frac{\partial^3 f(\bar{x})}{\partial xyz}h_1h_2h_3 + \ldots \end{aligned}$$

In [16] findet man Differenzensterne zur Annährung eindimensionaler Ausdrücke wie f_x, f_{yy}, f_{zzz} usw. Aus diesen können leicht Differenzensterne für partiellen Ableitungen in mehrere Richungen erzeugt werden. Alle verwendeten eindimensionalen Sterne sind aus [16].
Sei nun z.B. eine Annäherung von f_{xxy} gesucht. Wir finden sie, indem wir Differenzensterne von f_{xx} und f_y geeignet miteinander verknüpfen. Dies ist so zu verstehen: Es gelten

$$\frac{f(x-h,y) - 2f(x,y) + f(x+h,y)}{h^2} = f_{xx}(x,y) + \mathcal{O}(h^2)$$

und

$$\frac{-f(x,y-h) + f(x,y+h)}{2h} = f_y(x,y) + \mathcal{O}(h^2).$$

Diese beiden Annäherungen werden wie in folgenden Schema kombiniert (Das doppelt eingerahmte Gewicht entspricht der Stelle, in der angenähert werden soll):

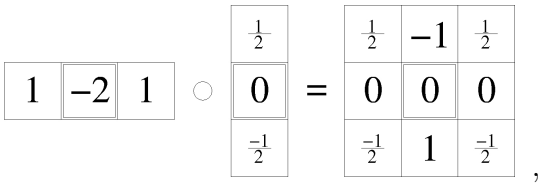


Abbildung 3.4: Graphische Herleitung von f_{xxy}

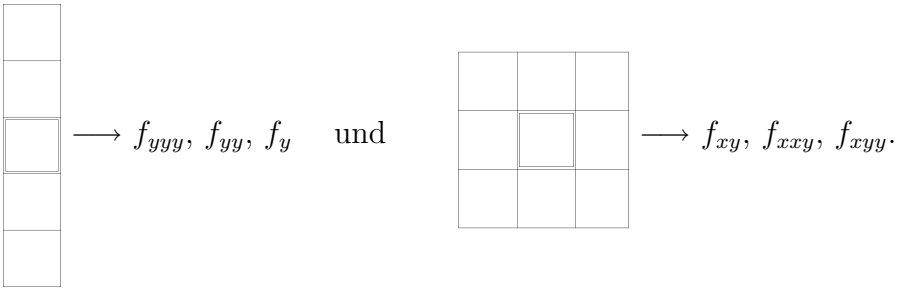
was als

$$\begin{aligned} f_{xxy} \approx & \frac{-f(x-h,y-h)+2f(x,y-h)-f(x+h,y-h)}{2h^3} \\ & +\frac{f(x-h,y+h)-2f(x,y+h)+f(x+h,y+h)}{2h^3} \end{aligned} \tag{3.20}$$

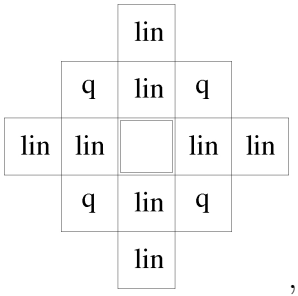
zu verstehen ist. Mittels Taylorentwicklung verifiziert man, daß durch die so erzeugte Formel eine Approximation mit einem Fehler von $\mathcal{O}(h^2)$ gegeben ist. Approximationen anderer Terme konstruiert man dann entsprechend. Es wurden immer Differenzensterne ausreichender Ordnung verwendet, so daß die jeweiligen Taylorreihen auch numerisch von der entsprechenden Ordnung approximiert wurden.
Jetzt kann geklärt werden, wie viele Auswertungen im $\mathbb{R}^n$ nötig sind, um $T_3(\bar{x}+h)$ durch Differenzensterne anzunähern. Betrachten wir hierfür zuerst die Ableitungen in eine Richtung, z.B. in Richung x. Wir benötigen einen Stern der Form

der uns die Approximation von f_x, f_{xx} und f_{xxx} ermöglicht. Dies stellen wir folgendermaßen dar:

Es seien weiterhin exemplarisch aufgeführt:



Betrachten wir nun den zweidimensionalen Stern der Form



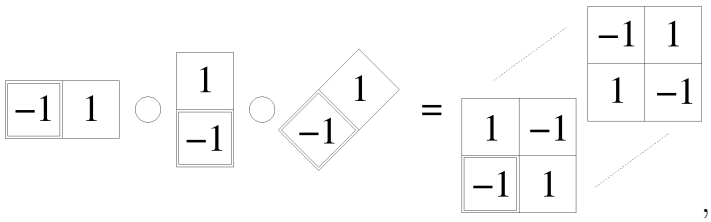
,

Abbildung 3.5: Zweidimensionaler Differenzenstern

der in der x_ix_j-Ebene liegt. Mit den Auswertungen in diesem Stern kann man in (3.3) fast alle Terme, die Abeitungen in den Richtungen x_i und x_j enthalten annähern. Ausnahme sind noch die Ableitungen der Form $f_{x_ix_jx_k}$ mit $x_i \neq x_k$ und $x_j \neq x_k$, auf die wir später eingehen. Vorher klären wir, wie viele Auswertungen für die Ableitungen benötigt werden, die man bisher annähern kann.
Wir gehen dazu vom $\mathbb{R}^n$ aus. Es gibt dort $\binom{n}{2}$ Kombinationen verschiedener Richtungen x_i, x_j, also benötigen wir $\binom{n}{2}$ Sterne mit je 13 Feldern wie in Abbildung 3.5. Das bedeutet aber nicht, daß wir dann $13\binom{n}{2}$ Auswertungen brauchen. Betrachtet man die Menge aller Sterne, so stellt man fest, daß sie alle das selbe Zentrum hat, dort also nur einmal ausgewertet werden muß. Es ist weiterhin leicht einzusehen, daß es insgesamt $4n$ mit „lin“ gekennzeichneten Knoten gibt. Die mit „q“ markierten Stellen dagegen kommen immer nur in einem Stern vor. Dies ergibt insgesamt einen Bedarf von

$$\#_2(n) = 1 + 4n + 4\binom{n}{2}$$

Auswertungen.
Für Ableitungen $f_{x_ix_jx_k}$ in drei verschiedene Richtungen setzen wir die zweifache Verkettung des einseitigen Differenzenquotienten von der Ordnung 1 an. Das Schema in den Richtungen x, y und z

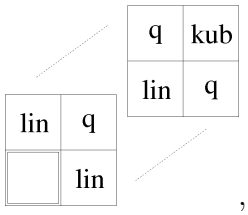


,

führt zu der Formel

$$\begin{aligned} f_{xyz} &= \frac{-f(x,y,z)+f(x+h,y,z)+f(x,y+h,z)-f(x+h,y+h,z)}{h^3} \\ &+\frac{f(x,y,z+h)-f(x+h,y,z+h)}{h^3} \\ &+\frac{-f(x,y+h,z+h)+f(x+h,y+h,z+h)}{h^3}+\mathcal{O}(h). \end{aligned}$$

Markieren wir die einzelnen Felder des dreidimesionalen Stern wieder wie vorher



,

so erkennt man, daß alle bis auf das mit „kub“ gekennzeichnete Feld bereits in einem der Sterne aus Abbildung 3.5 auftreten. Im $\mathbb{R}^n$ gibt es $\binom{n}{3}$ Tripel der Form (x_i, x_j, x_k) mit verschiedenen Richtungen, folglich kommen noch einmal $\binom{n}{3}$ Auswertungen zu $\#_2$ dazu. Wir können also $T_3(\bar{x}+h)$ mit

$$\#_3(n) = \binom{n}{3} + 4\binom{n}{2} + 4n + 1$$

Auswertungen annähern. Tabellarisch heißt das

n	1	2	3	4	5	6	7	8	9	10	11	12
$\#_3(n)$	5	13	26	45	71	105	148	201	265	341	430	533

Die Untersuchung von $T_4(\bar{x}+h)$ verläuft nach dem selben Prinzip, liefert

$$\#_4(n) = \binom{n}{4} + 4\binom{n}{3} + 6\binom{n}{2} + 4n + 1$$

und damit

n	1	2	3	4	5	6	7	8	9	10
$\#_4(n)$	5	15	35	70	126	210	330	495	715	1001

Für $n = 10$ wäre (3.7) gerade noch durchführbar. Das Halley-Verfahren dagegen kann ohne Probleme in die globale Suche eingebettet werden. Wahrscheinlich kann wie in Abschnitt 3.1 ein Verfahren der Ordnung 5 hergeleitet werden. Es ist allerdings zu erwarten, daß dies mehr Auswertungen benötigt als uns zur Verfügung stehen.

3.3 Numerische Resultate

Wir haben nun zwei Verfahren zur Verfügung, die zumindest in der Theorie einen Vorteil gegenüber dem Newton-Verfahren andeuten. Es wird nun geklärt, ob sich dies auch in die Praxis fortsetzt.
Vorher ist die Diskussion einiger Feinheiten notwendig. Es muß schließlich klar sein, was genau verglichen wird. So wurde bisher noch nicht darauf eingegangen, wie die Schrittweite der Verfahren gesteuert wird: Dies ist nötig, weil Newton- und Halley-Verfahren und auch $\mathcal{V}$ in ihren bisherigen Formulierungen nur dann einen Abstieg liefert, wenn man schon in der Nähe der Lösung ist. In unserem Fall käme nur eine gradientenfreie Steuerung im Stile der Goldstein-Armijo-Regel [36] in Frage, weil sonst zu viele Auswertungen verbraucht würden. Wir haben aber nicht die Möglichkeit verschiedene Schrittweiten nacheinander auf ihre Abstiegseigenschaft zu testen, sondern müssen alle in Frage kommenden Punkte apriori wählen. Deshalb wählen wir die minimale Schrittweite ε_s und definieren für den Punkt $x \in \mathbb{R}^n$ und die Suchrichtung $h \in \mathbb{R}^n$ die Suchpunkte

$$S(x,h) := \{x+(1/2)^k h \text{ mit } k \in \mathbb{N}_0, \|(1/2)^k h\| > \varepsilon_s \text{ und } x+(1/2)^k h \in \mathcal{D}\} \quad (3.21)$$

Sollte $x+h$ außerhalb von $\mathcal{D}$ liegen, so wird $\lambda_h \in \mathbb{R}$ so gewählt, daß $x+\lambda_h h \in \partial\mathcal{D}$ gilt, dieser Punkt auf dem Rand wird dann noch zu $S(x,h)$ dazugenommen. Liegt x auf $\partial\mathcal{D}$, so wird unterschieden zwischen den Suchrichtungen, die in $\mathcal{D}$ liegen und denen, die aus $\mathcal{D}$ herausführen. Für erstere gilt die analoge Definition von $S(x,h)$. Für die anderen wird $x+h$ auf $\mathcal{D}$ projeziert, was den Punkt $x+h_p$ liefert, und dann $S(x,h_p)$ wie in (3.21) ermittelt. Wir definieren dann noch

$$S_m(x,h) = \operatorname{argmin}\{f(\xi) \,|\, \xi \in S(x,h)\}, \quad (3.22)$$

was gleichzeitig die nächste Iterierte ist. Dies genügt dem Anpruch, in jedem Schritt einen möglichst großen Fortschritt hinsichtlich der Zielfunktionswerte zu

machen.
Auf dieser Grundlage werden nun die drei Verfahren miteinander verglichen. Üblicherweise testet man die Kandidaten an einem in der Forschung anerkannten Testsatz. Ein Auszug der gängigsten Benchmarkprobleme findet sich in [19], diese Auswahl wurde auch hier verwendet. Als zulässige Menge $\mathcal{D}$ wurde entprechend der Aufgabenstellung ein Quader gewählt. Nun sind die Testfunktionen nicht so strukturiert, daß sich das Minimum immer in $[0,1]^n$ befindet. Deswegen wurde $\mathcal{D}$ für jede Testfunktion individuell gewählt. Am Beispiel der Rosenbrock Funktion

$$R(x,y) = (1-x)^2 + 100(y-x^2)^2$$

wird nun erläutert, wie das Newton-Verfahren, das Halley-Verfahren und $\mathcal{V}$ miteinander verglichen wurden.
So wurde für $R(x,y)$ der Quader

$$\mathcal{D}_R := \{x \in \mathbb{R}^2 \,|\, a_i \leq x_i \leq b_i\}$$

mit $a = (-1,0)$ und $b = (3,2)$ definiert. Damit am Ende ein möglichst aussagekräftiges Ergebnis vorliegt, wurden in $\mathcal{D}_R$ viele verschiedene Startpunkte $x_s^1, \ldots, x_s^M$ zufällig ausgewählt. M variiert von Problem zu Problem und wurde jeweils so gewählt, daß die M Vergleiche in ca. 6 Stunden durchgeführt werden konnten.
Um $\mathcal{V}$ zu testen, wurde wie folgt vorgegangen: In x_s^1 wurde, wie in Abschnitt 3.2 beschrieben, ein Differenzenstern erzeugt und die Testfunktion auf diesem ausgewertet. Aus diesen Auswertungen wurde mit Hilfe der Theorie aus Abschnitt 3.2 die Suchrichtung $h_g := -\nabla(f(x_s^1))$ und die Suchrichtungen h_N, h_H und $h_\mathcal{V}$ des Halley- bzw. Newton-Verfahrens und von $\mathcal{V}$ errechnet. Damit sind die Suchrichtungen dieser drei Verfahren im Punkt x_s^1 gemeint. Für jede dieser vier Richtungen wurde $S_m(x_s^1, h_g), S_m(x_s^1, h_N), S_m(x_s^1, h_H)$ bzw. $S_m(x_s^1, h_\mathcal{V})$ gemäß (3.22) bestimmt. Ist

$$\text{argmin}\{f(S_m(x_s^1, h_g)), f(S_m(x_s^1, h_N)), f(S_m(x_s^1, h_H,)), f(S_m(x_s^1, h_\mathcal{V})\} \qquad (3.23)$$

weniger als ε vom Minimum entfernt, so brechen wir diese Suche ab und das Minimum gilt als nach einem Schritt gefunden. Falls nicht, liefert (3.23) das Zentrum für einen neuen Differenzenstern, wo wir wieder die vier Suchrichtungen generieren usw. Irgendwann ist das Minimum gefunden oder die Maximalschrittzahl n_{max} überschritten. Die Anzahl der Schritte, die bis zum Abbruch vorgenommen wurden, bezeichnen wir als $s_\mathcal{V}^1$. Damit wird angegeben, wie viele Schritte unter

Hinzunahme von $\mathcal{V}$ von x_s^1 aus benötigt wurden, um das Minimum zu finden.
Ist dies abgeschlossen, starten wir die Suche von x_s^1 aus erneut. Der Unterschied zum eben beschriebenen ist, daß nur noch jeweils die Suchrichtungen h_g, h_N und h_H ermittelt und überprüft werden. Ansonsten ist das Vorgehen identisch. Dadurch wird ermittelt, wie viele Schritte gebraucht werden, wenn das Halley-Verfahren Bestandteil der Methode ist. Die so ermittelte Schrittzahl wird mit s_H^1 bezeichnet.
Zuletzt wird noch einmal wie eben beschrieben verfahren, es gehen aber nur noch die Suchrichtungen h_g und h_N ein. Es wird also ermittelt wie viele Schritte eine Kombination aus Newton- und Gradientensuche benötigt, um von x_s^1 ins Minimum zu gelangen. Die Anzahl dieser Schritte wird mit s_N^1 bezeichnet.
Dieses Vorgehen wurde gewählt, weil es in der parallelen Problemstellung keine Rolle spielt, ob man $S(x, h)$ für eine, zwei oder drei Richtungen h bestimmt. In jedem Fall sind viel weniger als 1000 Auswertungen nötig. Die zweite eben beschriebene Methode wurde auch in das in Abschnitt 4.2 entwickelte Verfahren integriert.
Die drei eben beschriebenen Verfahren wurden von jedem $x_s^i,\ i = 1, \dots M$, aus gestartet. Die benötigten Schrittzahlen werden jeweils als $s_\mathcal{V}^i, s_H^i$ bzw. s_N^i bezeichnet, wodurch die Vektoren $s_\mathcal{V}, s_H$ und s_N entstehen.
Es sei angemerkt, daß obige Verfahren in keinem einzigen Testlauf bei keiner der Testfunktionen gegen einen anderen Punkt als ein lokales Minimum konvergiert sind.
Für $R(x, y)$ wurden mit $M = 2500$ die Vektoren $s_\mathcal{V}$, s_h und s_n ermittelt. Wir definieren

$$q_h^i := \frac{s_h^i}{s_n^i} \quad \text{bzw.} \quad q_\mathcal{V}^i := \frac{s_\mathcal{V}^i}{s_h^i},$$

wodurch angegeben wird, wie viel Aufwand das Halley-Verfahren im Vergleich zum Newton-Verfahren bzw. $\mathcal{V}$ im Vergleich zum Halley-Verfahren vom Startpunkt x_s^i aus macht. Interessant scheinen nun folgende Fragen zu sein: Wie verhalten sich

$$\varnothing_h := \frac{1}{M} \sum_{i=1}^{M} q_h^i \quad \text{bzw.} \quad \varnothing_\mathcal{V} := \frac{1}{M} \sum_{i=1}^{M} q_\mathcal{V}^i \quad ?$$

Dies gibt einen Überblick darüber, wie viel man durch Einsatz des Halley-Verfahrens bzw. $\mathcal{V}$ im Mittel einsparen kann, wenn man vom selben Punkt aus startet. Oder ob man evtl. sogar mehr Aufwand betreiben muß.
Weiterhin interessant ist die Frage, in wie vielen Fällen das Halley- im Vergleich zu Newton-Verfahren eine Verbesserung bringt, bzw. wie oft das Newton-Verfahren

besser war. Wie wahrscheinlich ist es, überhaupt etwas einzusparen bzw. zu verlieren. Es geht also um die Größen

$$h_b := \frac{\#\{i \mid q_h^i < 1\}}{M}, \quad h_g := \frac{\#\{i \mid q_h^i = 1\}}{M} \quad \text{und} \quad h_s := \frac{\#\{i \mid q_h^i > 1\}}{M}$$

bzw.

$$\mathcal{V}_b := \frac{\#\{i \mid q_\mathcal{V}^i < 1\}}{M}, \quad \mathcal{V}_g := \frac{\#\{i \mid q_\mathcal{V}^i = 1\}}{M} \quad \text{und} \quad \mathcal{V}_s := \frac{\#\{i \mid q_\mathcal{V}^i > 1\}}{M}.$$

Um Ausreißer aus der Statistik zu nehmen, wurden die 2.5% kleinsten q_h^i bzw. $q_\mathcal{V}^i$ und die 2.5% größten nicht beachtet.
Um zu untersuchen, ob sich die Konvergenz dritter bzw. vierter Ordnung nur in der Nähe des Minimums x^* auswirkt, wurde die Abbruchbedingung $\|x^i - x^*\| < \varepsilon$ für verschiedene ε getestet. Für $R(x, y)$ mit $\varepsilon = 0.01$ erhielten wir

$$\varnothing_h = 71.7\%, \quad h_b = 85.4\% \quad h_g = 9.4\% \quad h_s = 5.2\%.$$

Das heißt, daß man sich gegenüber dem Newton-Verfahren mit erheblicher Wahrscheinlichkeit verbessert. Im Mittel braucht man nur etwa 70% der Iterationen. Das Risiko, daß man schlechter abschneidet, ist mit ca. 5% durchaus gering.
In der Theorie ist $\mathcal{V}$ zwar noch schneller als das Halley-Verfahren, jedoch sind die numerischen Ergebnisse enttäuschend:

$$\varnothing_V = 99.7\%, \quad \mathcal{V}_b = 31.1\% \quad \mathcal{V}_g = 54\% \quad \mathcal{V}_s = 14.9\%.$$

Im Mittel hat man keinen ernsthaften Vorteil gegenüber dem Halley-Verfahren, auch die Chance, sich überhaupt zu verbessern, scheint relativ gering im Vergleich zu dem Aufwand, den man betreiben muß.
Für $\varepsilon = 0.001$ erhalten wir

$$\varnothing_h = 71.4\%, \quad h_b = 89.2\% \quad h_g = 6.5\% \quad h_s = 4.3\%.$$

bzw.

$$\varnothing_\mathcal{V} = 100.2\%, \quad \mathcal{V}_b = 24.2\% \quad \mathcal{V}_g = 60.9\% \quad \mathcal{V}_s = 14.9\%.$$

und $\varepsilon = 0.0001$ liefert

$$\varnothing_h = 70.4\%, \quad h_b = 92.8\% \quad h_g = 3.4\% \quad h_s = 3.8\%.$$

bzw.

$$\varnothing_\mathcal{V} = 99.9\%, \quad \mathcal{V}_b = 27.2\% \quad \mathcal{V}_g = 59.3\% \quad \mathcal{V}_s = 13.5\%.$$

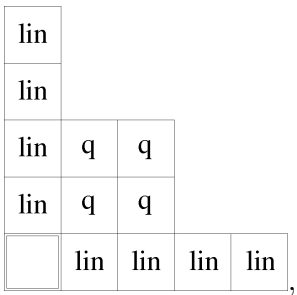
Dies paßt zur These, daß das Halley-Verfahren unabhängig vom Startpunkt schneller ist als das Newton-Verfahren und seinen Vorteil nicht nur aus den letzten Iterationen nahe am Minimum erzielt. Schon vorher hat man einen Vorteil.
Sehr ähnliche Ergebnisse erhielt man für die übrigen Testfunktionen:

Funktion	ε	$\varnothing_h$	h_b	h_g	h_s	$\varnothing_{\mathcal{V}}$	$\mathcal{V}_b$	$\mathcal{V}_g$	$\mathcal{V}_s$
	0.01	71.7	85.4	9.4	5.2	99.7	31.1	54	14.9
Rosenbrock	0.001	71.4	89.2	6.5	4.3	100.2	24.2	60.9	
	0.0001	70.4	92.9	3.4	3.8	99.9	27.2	59.3	13.5
	0.01	78.8	93	7	0	98.6	13.3	82.6	4.1
Helix	0.001	78	96.2	4.8	0	99	12.6	81.8	5.6
	0.0001	77.5	97.8	2.2	0	98.6	13.9	82.2	3.9
	0.01	88.8	57.4	41.8	0.8	97.1	16	84	0
Gauß	0.001	86.9	70.1	29.3	0	96.5	20.7	79.3	0
	0.0001	85.2	82.3	17.7	0	96.2	22.9	77.1	0
	0.01	84.8	100	0	0	92.6	79.7	6.2	14.1
Powell s.k.	0.001	84.2	100	0	0	92.6	80.6	4.2	15.2
	0.0001	83.1	100	0	0	94.2	79.7	5.3	15
	0.01	93.2	36.1	63.2	7.7	99.6	3.3	97.7	0
Box	0.001	92.4	40.1	59.5	0.4	99.5	4.5	95.5	0
	0.0001	91.6	45.4	53.9	0.7	99.4	5.2	94.1	0.7
	0.01	100.4	11.6	77.6	10.8	100	0	100	0
Beliebig dim.	0.001	97.4	21.9	70.1	7.3	100	0	100	0
	0.0001	98.5	19.3	71.2	9.5	99.5	3.1	96.9	0
	0.01	77.3	95.8	4.2	0	100	1.1	97.5	1.4
Watson	0.001	77	96	4	0	99.3	6	94	0
	0.0001	76.3	98.2	1.8	0	98.3	15.1	84.9	0
	0.01	86.3	98.4	1.6	0	97.7	46.1	30.3	23.6
Penalty I	0.001	84.7	100.0	0	0	97.5	52.8	27.4	19.8
	0.0001	82.4	100.0	0	0	97.5	46.6	30.6	22.8
	0.01	89.1	59.5	40.5	0	96.5	18.7	80.8	0.5
Penalty II	0.001	85.5	81.5	18.5	0	95.6	25.1	74.5	0.4
	0.0001	84.7	87.4	12.6	0	94.2	33.7	66.2	0
	0.01	36.3	89.5	10.5	0	81.8	41.0	56.0	3.0
Brown s.k.	0.001	37.6	89.4	10.6	0	81.9	42.1	55.6	2.2
	0.0001	37.8	90.3	9.7	0	82.4	42.4	54.4	3.2

Funktion	ε	$\varnothing_h$	h_b	h_g	h_s	$\varnothing_\mathcal{V}$	$\mathcal{V}_b$	$\mathcal{V}_g$	$\mathcal{V}_s$
	0.01	75.9	100.0	0	0	96.9	17.8	82.2	0
B. Dennis	0.001	75.2	100.0	0	0	97.0	18.0	82.0	0
	0.0001	74.1	100.0	0	0	97.0	18.2	81.8	0
	0.01	81.9	86.0	0.4	13.6	104.7	39.6	17.4	43.0
Cox	0.001	82.7	88.1	1.0	10.9	105.0	42.8	17.1	40.1
	0.0001	83.4	88.1	0.8	11.1	104.4	41.1	19.0	39.9
	0.01	89.3	51.7	48.3	0	96.2	21.2	75.5	3.2
Trig. F.	0.001	87.5	66.2	33.8	0	93.0	36.0	63.0	1.0
	0.0001	85.2	78.1	21.9	0	93.2	36.6	61.6	1.8
	0.01	74.7	100.0	0	0	91.8	72.7	27.3	0
Powell Sing.	0.001	72.8	100.0	0	0	90.9	87.3	12.7	0
	0.0001	73.5	100.0	0	0	92.3	79.9	19.0	1.1
	0.01	85.1	75.9	24.1	0	95.8	25.4	72.4	2.2
Beale	0.001	83.1	86.8	13.2	0	94.2	35.6	63.0	1.4
	0.0001	82.2	90.9	9.1	0	93.9	39.0	59.8	1.3
	0.01	79.1	94.1	2.1	3.8	96.7	41.2	50.5	8.3
Wood	0.001	78.8	95.6	1.4	3.1	96.5	44.5	47.5	8.0
	0.0001	78.5	96.2	1.3	2.5	96.0	49.9	45.4	4.6

3.4 Differenzensterne auf dem Rand

Die in Abschnitt 3.2 hergeleiteten Differenzensterne machen nur im Inneren des Suchgebietes Sinn. Auch wenn die Suche auf dem Rand in dieser Arbeit keine Rolle spielt, kann dennoch die Stuation auftreten, daß in einem Punkt auf dem Rand approximiert werden muß. Mit dem bisherigen Ansatz würden die Auswertungen teilweise außerhalb der zulässigen Menge liegen.
Dies kann vermieden werden. Analog zu Abschnitt 3.2 betrachtet man einen zweidimensionalen Stern der Form



,

der nun aber eher symbolisch zu verstehen ist: Es sind nicht zwei aufeinander senkrecht stehende Richtungen x_i und x_j gemeint, sondern zwei beliebige linear unabhängige normierte Richtungen d und r. Es handelt sich also um einen „schiefen“ Stern. Es würde sich dann beispielsweise analog zur Herleitung von f_{xxy} nach Abbildung 3.2 und (3.20)

$$\begin{aligned} f_{ddr} \approx & \frac{-f\left(x-h(d_1+r_1), y-h(d_2+r_2)\right)+2f(x-hr_1, y-hr_2)}{2h^3} \\ & + \frac{-f(x+h(d_1-r_1), y+h(d_2-r_2))}{2h^3} \\ & + \frac{f\left(x+h(-d_1+r_1), y+h(-d_2+r_2)\right)-2f(x+hr_1, y+hr_2)}{2h^3} \\ & + \frac{f(x+h(d_1+r_1), y+h(d_2+r_2))}{2h^3} \end{aligned}$$

ergeben. Hat man nun n linear unabhängige normierte Richtungen $r_1, \ldots, r_n$ im $\mathbb{R}^n$, die vom Auswertungspunkt ins Innere des Suchgebiets zeigen, so kann man daraus alle partiellen ersten und zweiten Ableitungen der Form f_{r_i} bzw. $f_{r_j r_k}$ erzeugen.
Es gilt

$$f_{r_i} = \langle \nabla f, r_i \rangle,$$

daraus generiert man das Lineare Gleichungssystem

$$\begin{pmatrix} r_1^T \\ \vdots \\ r_n^T \end{pmatrix} \nabla f = \begin{pmatrix} f_{r_1} \\ \vdots \\ f_{r_n} \end{pmatrix},$$

das sich eindeutig nach ∇f auflösen läßt, weil die Systematrix wegen der linearen Unabhängigkeit der Richtungen $r_1, \ldots, r_n$ vollen Rang hat.
Die Berechnung der Hessematrix ist etwas komplizierter, aber möglich. Zuerst betrachten wir die beliebige Koordinatenrichtung x_i und die beliebige normierte Richtung d. Dann gilt

$$\begin{aligned} f_{x_i d} &= \frac{\partial f_{x_i}}{\partial d} = \langle \nabla f_{x_i}, d \rangle = \sum_{j=1}^{n} f_{x_i x_j} d_j = \frac{\partial}{\partial x_i} \sum_{j=1}^{n} f_{x_j} d_j \\ &= \frac{\partial}{\partial x_i} \langle \nabla f, d \rangle = \frac{\partial f_d}{\partial x_i} = f_{d x_i}. \end{aligned} \tag{3.24}$$

Mit diesem Wissen generieren wir aus

$$\frac{\partial f_{x_j}}{\partial r_i} = \langle \nabla f_{x_j}, r_i \rangle$$

die n Linearen Gleichungssyteme

$$\begin{pmatrix} r_1^T \\ \vdots \\ r_n^T \end{pmatrix} \nabla f_{x_j} = \begin{pmatrix} \frac{\partial f_{x_j}}{\partial r_1} \\ \vdots \\ \frac{\partial f_{x_j}}{\partial r_n} \end{pmatrix} = \begin{pmatrix} f_{x_j r_1} \\ \vdots \\ f_{x_j r_n} \end{pmatrix} \quad \text{mit} \quad j = 1, \ldots, n.$$

Die Lösung ∇f_{x_j} eines solchen Systems ist gerade die $j-$te Spalte der Hessematrix. In der rechten Seite sind jedoch noch Terme enthalten, die wir so noch nicht mit dem schiefen Stern approximieren können. Betrachten wir deshalb eine beliebige der Richtungen, z.B. r_i. Wir haben

$$\nabla f_{r_i} = (f_{r_i x_1}, \ldots, f_{r_i x_n}) = (f_{x_1 r_i}, \ldots, f_{x_n r_i}) \,.$$

Können wir ∇f_{r_i} bestimmen, dann haben wir auch die rechten Seiten der Linearen Gleichungssysteme. Für die ∇f_{r_i} gilt aber gerade

$$f_{r_i r_k} = \langle \nabla f_{r_i}, r_k \rangle \quad \Rightarrow \quad \begin{pmatrix} r_1^T \\ \vdots \\ r_n^T \end{pmatrix} \nabla f_{r_i} = \begin{pmatrix} f_{r_i r_1} \\ \vdots \\ f_{r_i r_n} \end{pmatrix}.$$

Auf diese Art kann man mit Hilfe eines „schiefen“ Sterns Gradient und Hessematrix erzeugen. Weil eine Suche auf dem Rand in der Anwendung kaum auftritt, wurde dieses Vorgehen nicht auf den dreidimesionalen Stern erweitert.

Kapitel 4

Neue Verfahren zur Optimierung teurer Funktionen

In die Entwicklung des im Folgenden vorgestellten Verfahrens gingen folgende Grundgedanken ein: Im Gegensatz zu den beiden erfolgreichsten Verfahren, CORS und RBF, soll klarer zwischen lokaler und globaler Suche unterschieden werden. Dort werden jeweils Zyklen durchlaufen. So wird z.B. zu Beginn eines Zyklus mehr global gesucht. Das heißt, es wird eher in einem entlegenen Punkt ausgewertet. Am Ende des Zyklus wird dann nahe des globalen Minimums eines Interpolanden ausgewertet. In den Schritten dazwischen findet ein Übergang zwischen dem Auffüllen und der lokalen Suche statt.
Auch in unserem Verfahren gibt es eine globale und eine lokale Komponente. Allerdings sind sie klar voneinander getrennt, wobei in beide Bestandteile etwa die selbe Anzahl an Auswertungen investiert wird. Es gibt keine dazwischen gelagerten Schritt, sondern es wird abwechselnd rein global und dann wieder rein lokal gesucht.
Die globale Suche wird nach dem Prinzip eines einstufigen Verfahrens erfolgen. Dabei soll die Bewertung einer möglichen Auswertungsstelle auf ihre Interessantheit hin geometrisch interpretierbar und formal leicht beschreibbar sein. Die lokale Suche soll nach Möglichkeit nicht mehr als die bisherigen Auswertungen zu seiner Durchführbarkeit benötigen. All dies führte dazu, daß das in dieser Arbeit entwickelte neue Verfahren auf Techniken aus den Abschnitten 2.2 und 2.4 zurückgreift. Man kann zwei verschiedene Versionen formulieren: Eine sequentielle und eine parallele. Ursprünglich war nicht geplant, eine sequentielle Fassung anzugeben. Jedoch waren die numerischen Ergebnisse mit denen der bisher bekannten Methoden konkurrenzfähig, weshalb auch diese Version vorgestellt wird. Das parallele Verfahren läßt sich schön aus dem sequentiellen herleiten, deswegen

wird dieses zuerst hergeleitet.

4.1 Sequentielles Verfahren

Zu Beginn wird der globale Bestandteil des Algorithmus erläutert. Dazu wird vorausgesetzt, daß schon $m+1$ Knoten $\Xi = \{x_0, \ldots, x_m\}$ und dort die Zielfunktionswerte Y bekannt sind. Anwender sind meist daran interessiert, den bisher besten Zielfunktionswert (z.B. Kosten) y_{min} mindestens um einen bestimmten Prozentsatz zu unterbieten. Hier wird das Ziel verfolgt, y_{min} um ca. 10% zu verbessern. Der Algorithmus funktioniert aber auch mit anderen Vorgaben. Mit $y_{max} := \max_{y_i \in Y} y_i$ setzen wir die Zielvorgabe

$$y^z := \begin{cases} 1.1 y_{min} & \text{falls } y_{min} < 0 \\ y_{min} - 0.1(y_{max} - y_{min}) & \text{falls } y_{min} = 0 \\ \min\{0.9 y_{min}, y_{min} - 0.1(y_{max} - y_{min})\} & \text{falls } y_{min} > 0 \end{cases} \tag{4.1}$$

Die für $y_{min} = 0$ gewählte Schranke ist sicher die plausibelste, kann aber nicht auf den Fall $y_{min} > 0$ übertragen werden, weil sonst die Konvergenz der entwickelten Verfahren nicht mehr garantiert werden kann. Nun wird die nächste Iterierte $x_{m+1} \in \mathcal{D}$ gesucht. Sie soll so gewählt werden, daß (x_{m+1}, y^z) möglichst gut zu Ξ, Y paßt. Wo also ist es am plausibelsten, daß der Wert y^z erreicht wird?
Wir müssen dazu eine sinvolle Kennzahl entwickeln, die dies beschreibt. Dazu wählen wir zunächst irgendeine zusätzliche Stelle $x^z \in \mathcal{D}$ und bestimmen eine Funktion Φ^z, die $(\Xi, x^z), (Y, y^z)$ interpoliert. Wir entscheiden uns für einen Ansatz nach dem Kalkül der Radialen Basisfuktionen. Es wird wie in (1.6) vorgegangen, mit

$$\phi(d) = d^3 \tag{4.2}$$

und linearem Polynom. Wir erhalten also

$$\Phi^z(x) = \sum_{i=0}^{m} \lambda_i \|x - x_i\|_2^3 + \lambda^z \|x - x^z\|_2^3 + a^T x + b, \tag{4.3}$$

wobei $a \in \mathbb{R}^n$ und $\lambda_i, \lambda^z, b \in \mathbb{R}$. Φ^z ist zweimal stetig differenzierbar, wie die Rechnung am Ende von Unterabschnitt 4.2.3 zeigt. Die gilt für die Ansätze $\phi(d) = d$ und $\phi(d) = d^2 \log(d)$ nicht, weshalb sie nicht berücksichtigt wurden. Alle auftretenden Koeffizienten werden aus dem Linearen Gleichungssystem, das wie (1.9)

entsteht, bestimmt. Φ^z ist in $\mathbb{R}^n$ zweimal stetig differenzierbar. Dies werden wir später sehen, wenn die Ableitungen konkret berechnet werden. Diese wurden auch in der Implementierung des Verfahrens verwendet. Die Hessematrix $H_{\Phi^z}(x)$ ist also wohldefiniert und das Quadrat ihrer Frobeniusnorm lautet

$$\|H_{\Phi^z}(x)\|_F^2 = \sum_{i,j=1}^{n} \left(H_{\Phi^z}(x)\right)_{i,j}^2 = \sum_{i,j=1}^{n} \left(\Phi^z_{\xi_i,\xi_j}(x)\right)^2 \tag{4.4}$$

In die Einschätzung einer potentiellen Auswertungsstelle x^z auf Ihre Interessantheit hin fließt nun folgende Größe ein,

$$\int_{\mathcal{D}} \|H_{\Phi^z}(x)\|_F^2 \, dx, \tag{4.5}$$

die eine schöne Deutung im Ein- und Zweidimensionalen besitzt: Im Eindimensionalen wird die Biegeenergie eines extrem dünnen und biegsamen Metallbandes gemessen, das durch die Knoten gelegt wurde, in 2-d handelt es sich um eine entsprechende Membran. Es ist plausibel, daß ein x^z besonders interessant ist, wenn man durch seine Hinzunahme das Band bzw. die Platte möglichst wenig verbiegen muß.
Diese Idee wurde in den betrachteten Beispielen, dort haben wir $\mathcal{D} = [0,1]^n$, noch um eine Barrierefunktion

$$B(x^z, \Xi) = -\frac{1}{2n\lfloor \#\Xi/2^n \rfloor} \sum_{j=1}^{n} (\ln(x_j^z) + \ln(1 - x_j^z)) \tag{4.6}$$

ergänzt, die automatisch mit wachsender Iterationzahl herunterfährt. Für eine positive reelle Zahl r ist $\lfloor r \rfloor$ die größte natürliche Zahl, die kleiner gleich r ist. Durch Null kann dabei nicht geteilt werden, weil man in den Testbeispielen zu Beginn immer schon 2^n Auswertungsdaten hatte.
Dies wurde eingeführt, weil im Gegensatz zum Eindimensionalen nicht auf dem gesamten Rand ausgewertet werden kann. Die durch (4.5) berechnete Krümmung fällt dann zu klein aus für x^z in Randnähe, weil sich der Interpoland nicht mehr „zurückwinden “ muß. Für beliebige Suchgebiete $\mathcal{D}$ muß man dann eine entsprechende, sinnvolle Barriere wählen. Wie in der restringierten Optimierung wird die Barriere im Laufe des Verfahrens heruntergefahren, so daß auch Randpunkte berücksichtigt werden, wenn sie interessant sind. Ohne diese Barriere neigt das Verfahren dazu im Mehrdimensionalen zuerst den Rand aufzufüllen, fast unabhängig davon, wie sich die Zielfunktion verhält.

Aus diesen beiden Bestandteilen setzt sich nun die Interessantheit

$$I(x^z) := \int_{\mathcal{D}} \|H_{\Phi^z}(x)\|_F^2 \, dx + B(x^z, \Xi) \tag{4.7}$$

der Stelle x^z zusammen. Somit wird die nächste Iterierte x_{m+1} gewählt nach

$$x_{m+1} := \underset{x^z \in D}{\operatorname{argmin}}(I(x^z)). \tag{4.8}$$

Die Berechnung von (4.7) und (4.8) ist nicht trivial. In Abschnitt 4.3 wird erläutert, wie sie implementiert wurde.

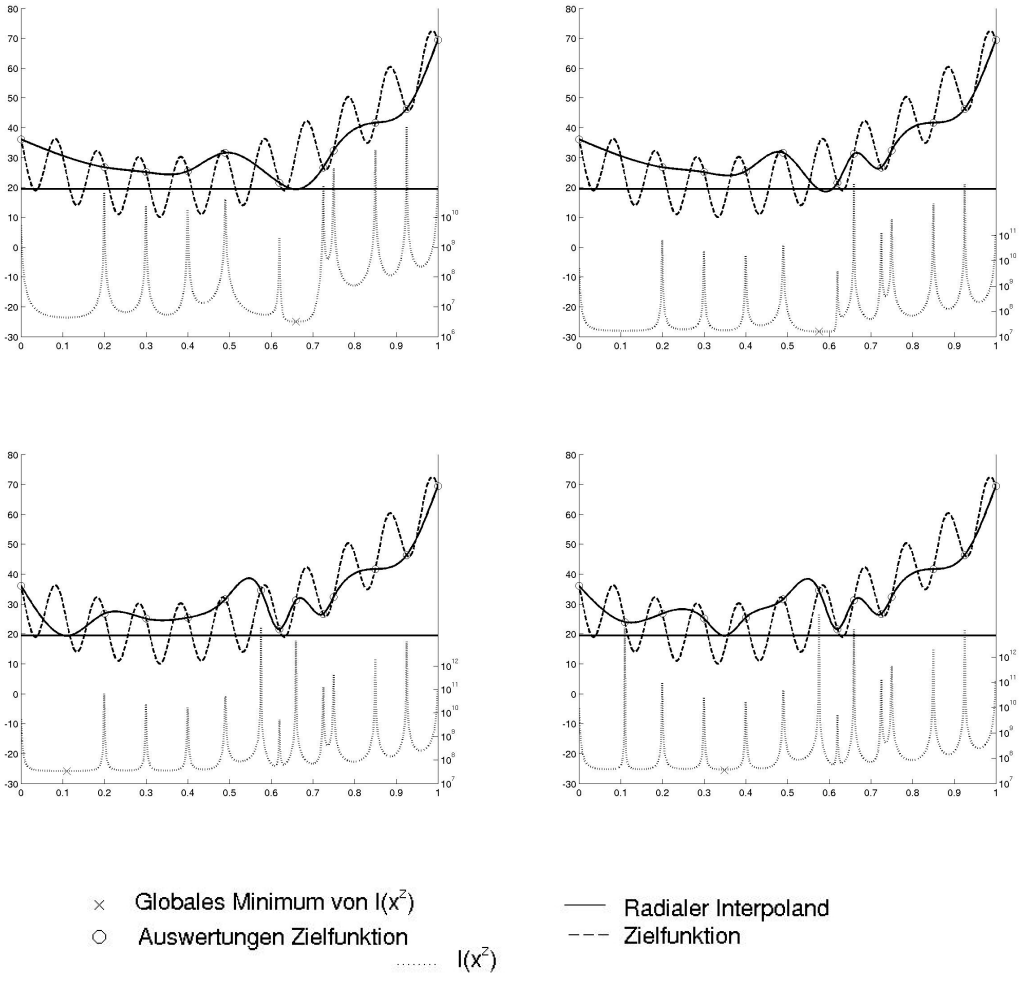


Abbildung 4.1: Minimierung nach (4.8)

Abbildung 4.1 zeigt die ersten vier Iterationen des eben beschriebenen Verfahrens, angewendet auf das Beispiel, an dem auch die Verfahren in Kapitel 2 getestet wurden.
Wurde die vorhergehenden Iterierte durch Minimierung von $I(x^z)$ ermittelt, so wird nun lokal gesucht. Wir machen dies, indem wir die bisherigen Knoten interpolieren und diesen Interpolanden global minimieren: Das globale Minimum des Interpolanden wird also als lokales Minimum der Zielfunktion interpretiert. Dies entspricht der Durchführung eines Schrittes von Algorithmus 2.
Der nächste Auswertungspunkt wird also abwechselnd einmal nach (4.8) und dann einmal durch globale Optimierung des Interpolanden ermittelt.
Wir bauen aber noch eine Erweiterung in Algorithmus 2 ein, um zu vermeiden, daß zu viele Auswertungen im selben lokalen Minimum verschwendet werden.
So werden abhängig von der Dimension n des Problems die letzten $2n$ *lokalen* Suchpunkte $x_i, \ldots, x_{i-2n+1}$ betrachtet: Wurde durch diese Suchpunkte keine signifikante Verbesserung erzielt, gilt also

$$\frac{|\min\{f(x_i), \ldots, f(x_{i-2n+1})\} - f(x_{i-2n})|}{|f(x_{i-2n})|} < \varepsilon,$$

so wird $\operatorname{argmin}\{f(x_i), \ldots, f(x_{i-2})\}$ als lokales Minimum betrachtet und dort nicht mehr lokal gesucht. Diese Vorgabe wird dadurch umgesetzt, daß das Newton-Verfahren in allen solchen lokalen Minima gestartet und auf den Radialen Interpolanden angewendet wird. Es wird dann im besten lokalen Minimum des Radialen Interpolanden ausgewertet, der nicht das Ziel einer solchen Newtonsuche war. Wir haben $\varepsilon = 0.01$ gewählt.
Um die algorithmische Beschreibung übersichtlich zu halten, wird dies nicht in die folgende Beschreibung aufgenommen, es wurde aber so implementiert. Resultat ist das sequentielle globale Optimierungsverfahren „GloSe“. Dieses Verfahren unterscheidet sich am meisten dadurch von den bisherigen Methoden, daß lokale und globale Suche klar getrennt sind und keine dazwischen gelagerten Schritte durchgeführt werden. Ein weiterer Unterschied ist, daß in die globale Suche extrem viel Rechenzeit investiert wird, siehe dazu Abschnitt 4.3. Während CORS und RBF mit vergleichbar wenig Aufwand auskommen, ist die Ermittlung von (4.8) sehr rechen- und damit zeitintensiv, weshalb sie nur bei extrem aufwendigen Zielfunktionen in Frage kommt. Im folgenden Abschnitt wird betrachtet, ob sich dieser Aufwand lohnt.

Algorithmus 7 GloSe

Input: Ξ,Y **Output:** x

1: $i := 0$
2: Setze $\Xi_0 := \Xi$ und $Y_0 := Y$
3: **while** Abbruchkriterium nicht erfüllt **do**
4: **if** Letzte Iterierte durch (4.8) bestimmt **then**
5: Setze $x :=$ Algorithmus 2 $(\Xi_i, Y_i, f, 1)$
6: **else**
7: Bestimme x nach (4.8)
8: **end if**
9: Setze $\Xi_{i+1} := \Xi_i \cup x$ und $Y_{i+1} := Y_i \cup f(x)$
10: i:=i+1
11: **end while**

4.1.1 Numerische Ergebnisse GloSe

Die konkurrierenden Verfahren wurden alle an den gleichen Funktionen getestet. Der Vergleichbarkeit wegen übernehmen wir diesen Testsatz von Dixon und Szegö, der zum erstenmal in [14] vorgestellt wird. Es handelt sich im wesentlichen um vier verschiedene Funktionen, von denen zwei noch als Abwandlungen enthalten sind. Im Anhang sind die genauen Funktionsdefinitionen und wesentliche Eigenschaften aufgeführt. Zum besseren Verständnis seien hier die Höhenlinien im untersuchten Gebiet angegeben. Die höherdimensionalen Funktionen wurden ohne Veränderung der Struktur zweidimensional dargestellt. Man sieht im Fall der Shekel Funktion den Plot zu $S5_2$ die Hartmann Funktion wurde durch $H3_2$ visualisiert. Die Funktionsdefinitionen findet man im Anhang. Die Goldstein-Price Funktion wurde logarithmisch geplottet, weil die Graphik sonst aufgrund der großen Unterschiede der Funktionswerte wertlos ist.
Wie in der Literatur üblich, wurde Algorithmus 7 durchgeführt, bis die Abbruchbedingung $|f_{min} - f^*|/f^* < 0.01$ erfüllt war. Die nachfolgenden Tabellen zeigen, wie viele Auswertungen GloSe dafür im Vergleich zu den bekannten Methoden benötigte. Besonders auffallend ist, daß GloSe bei jeder Funktion mit dem bisherigen Spitzenreiter (jeweils eingerahmt) mithalten kann und manchmal sogar noch etwas besser ist. Alle anderen Verfahren haben den Nachteil, nur für einen Teil der Funktionen sehr gut zu funktionieren, während sie bei anderen Funktionen deutlich abfallen. Insofern kann man behaupten, daß unser Verfahren für den üblichen Testsatz zufriedenstellend funktioniert.

Testfunktion	GloSe	CORS-RBF (SP1)	CORS-RBF (SP2)
Branin	24	34	40
Goldstein-Price	28	49	64
Hartmann3	24	25	61
Hartmann6	30	108	104
Shekel5	44	**41**	52
Shekel7	32	**46**	64
Shekel10	36	**51**	64

Testfunktion	GloSe	RBF-G	rbfSolve	DIRECT	EGO
Branin	24	44	**26**	63	28
Goldstein-Price	28	63	**27**	101	32
Hartmann3	24	43	**22**	83	35
Hartmann6	30	112	**87**	213	-
Shekel5	44	76	96	103	-
Shekel7	32	76	72	97	-
Shekel10	36	**51**	76	97	121

4.2 Paralleles Verfahren

Dadurch, daß wir in der parallelen Version der Aufgabenstellung viele Punkte gleichzeitig und ohne Mehraufwand setzen dürfen, ergeben sich neue Möglichkeiten. Trotzdem bleiben einige Gemeinsamkeiten zum eben beschriebenem Ansatz: Lokale und globale Strategien werden eingehen, die möglichst voneinander getrennt arbeiten sollen. Für beide wollen wir etwa gleich viele Auswertungen zur Verfügung stellen. Die globale Taktik soll auf der aus Abschnitt 4.1 aufbauen und diese dann verallgemeinern. Der lokale Sucher aus Abschnitt 2.2 wird zwar erneut verwendet, wir können uns nun aber lokale Suche in mehreren Bereichen erlauben. In besonders aussichtsreichen Knoten soll eine lokale Suche entsprechend dem Halley-Verfahren gestartet werden. Diese drei Komponenten sind zwar unabhängig voneinander zu betrachten, im Verfahrensverlauf werden sie dennoch teilweise aufeinander abgestimmt um keine Auswertungen zu verschwenden.

4.2.1 Globale Suche

Der globale Teil wird wieder von $\Xi = \{x_0, \ldots, x_m\}$ und dort bekanntem Y ausgehend konstruiert. Wir wollen nun m_g Auswertungen möglichst gewinnbringend

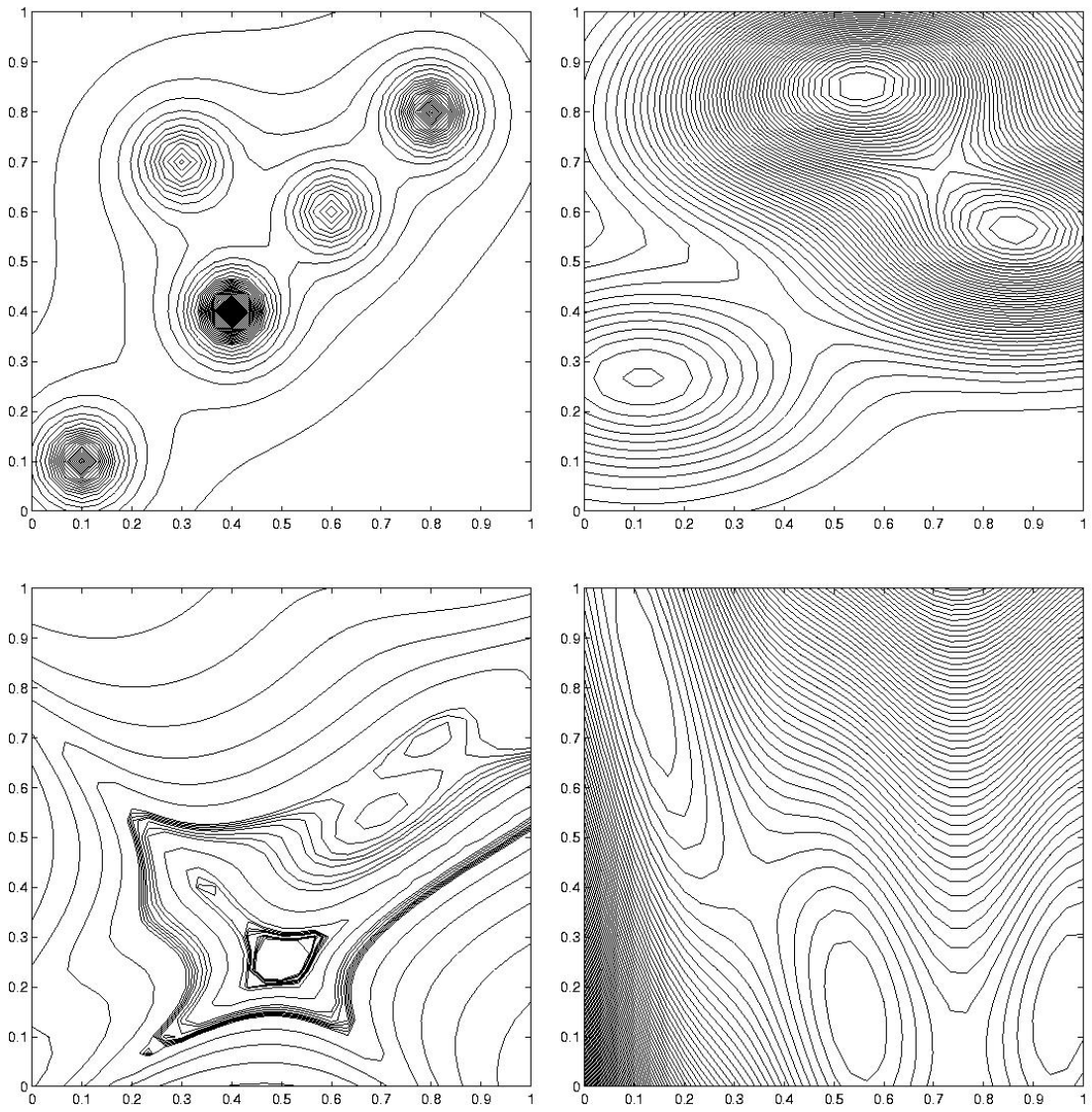


Abbildung 4.2: Shekel-, Hartmann-, Goldstein-Price- und Braninfunktion, zeilenweise von oben links nach unten rechts

verteilen. Dazu wird erst y^z wie in (4.1) festgelegt. Dieser Wert bleibt bei der Bestimmung aller m_g neuen Knoten gleich. Es folgt die Interpolation von Ξ, Y durch Φ gebildet nach (4.2) und (4.3), auch Φ wird in diesem Schritt nicht verändert. Grob gesprochen werden die Auswertungsstellen dadurch gefunden, daß man f durch Φ ersetzt und 4.8 wiederholt anwendet.
Genauer heißt dies: Der erste Knoten wird genauso wie in Abschnitt 4.1 ermittelt, wir nennen ihn x_1^g. An dieser Stelle dürfen wir nicht einfach wieder die Zielfunktion auswerten, haben aber noch $m_g - 1$ Knoten zu verteilen. Wir haben neben den Auswertungen nur durch Φ Informationen über den Verlauf der Zielfunktion. Daraus entstand die Idee, Ξ um x_1^g und Y um $\Phi(x_1^g)$ zu ergänzen, was dann Ξ_1 bzw Y_1 genannt wird. Per Definition ist Φ Interpoland auf Ξ_1, Y_1. Es wird nun (4.8) auf Ξ_1, Y_1 angewendet, was x_2^g liefert. Damit definieren wir analog Ξ_2, Y_2 und fahren fort. Diese Idee führt zu Algorithmus 8.

Algorithmus 8 GloPa$_{glob}$
Input : Ξ,Y,m_g, **Output:** $x_1, \ldots, x_{m_g}$

1: Definiere $\Xi_0 := \Xi$ und $Y_0 := Y$
2: Bestimme Interpolanden Φ der Form (4.2) und (4.3) für Ξ und Y
3: **for** $1 \leq i \leq m_g$ **do**
4: Berechne x_i nach (4.8)
5: Definiere $\Xi_i := \Xi_{i-1} \cup x_i$ und $Y_i := Y_{i-1} \cup \Phi(x_i)$
6: **end for**

4.2.2 Lokale Suche

Wie bereits angedeutet haben wir in der parallelen Problemstellung die Möglichkeit, mehrere lokale Suchen gleichzeitig durchzuführen. Im hier vorgestellten Verfahren wurde zum einen durch Auswertung in den lokalen Minima des Interpolanden, zum anderen mit dem Halley-Verfahren optimiert. Es folgt nun eine Beschreibung der Umsetzung. Dabei werden zunächst die einzelnen Bestandteile in Worten beschrieben. Darauf aufbauend wird eine algorithmische Darstellung gegeben, die möglichst übersichtlich gehalten wurde.
Zunächst wird die lokale Suche mit dem Halley-Verfahren beschrieben. Wir nennen sie im Folgenden Suche 1. Es werden folgende Punkte abgearbeitet:

„Bestimme $D_1(x)$“: Angenommen, wir haben einen Punkt x, in dem ein Halleyschritt durchgeführt werden soll. Dann bestimmen wir den zum Punkt x gehörigen Differenzenstern D_1 wie in Abschnitt 3.2.

„Bestimme $r_1(x)$“: Wurden durch eine hochparallele Auswertung die Zielfunkti-

onswerte auf D_1 ermittelt, so wird aus diesen die Suchrichtung $h_g := -\nabla(f(x))$ und die Suchrichtungen h_N bzw. h_H des Halley- bzw. Newton-Verfahrens errechnet. Für jede dieser drei Richtungen wird $S(x, h_g), S(x, h_N)$ bzw. $S(x, h_H)$ gemäß (3.21) bestimmt.

In einer hochparallelen Auswertung werden die Zielfunktionswerte auf $S(x, h_g)$, $S(x, h_N)$ und $S(x, h_H)$ ermittelt.
Da wir hochparallel auswerten dürfen, können wir gleichzeitig zur Suche 1 auch noch eine zweite Halleysuche, die Suche 2, laufen lassen. Angenommen, Suche 2 läuft schon und es liegt ein Differenzenstern $D_2(\tilde{x})$ samt Zielfunktionswerten auf $D_2(\tilde{x})$ vor. Dann können wir gleichzeitig zu „Bestimme $D_1(x)$“ auch noch die Suchrichtungen zu Suche 2 ermitteln, also „Bestimme $r_2(\tilde{x})$“ durchführen.
Analog dazu können wir gleichzeitig zu „Bestimme $r_1(x)$“ auch noch den Differenzenstern zu Suche 2 ermitteln, also „Bestimme $D_2(\tilde{x})$“ durchführen.
In der ersten hochparallelen Auswertung wird nur der Stern zu Suche 1 ermittelt, die Richtungen von Suche 2 können nicht berechnet werden, weil noch kein Stern für diese Suche vorliegt. Der erste Stern der Suche 1 wird in den Knoten mit dem kleinsten Zielfunktionswert gelegt.
Die beiden Suchen werden natürlich so aufeinander abgestimmt, daß sie möglichst nicht in das selbe und nicht in evtl. bereits vorhandene lokale Minima der Zielfunktion laufen. Dazu wird abgeschätzt, in welchen Zielpunkt Suche 1 und Suche 2 voraussichtlich laufen werden. Dies wurde wie folgt vorgenommen: Angenommen, es soll ein Differenzenstern für Suche 1 gesetzt werden. Es wird dann durchgeführt

„x =Finde Zentrum S_1“: Starte eine Newtonsuche für den Interpolanden im Zentrum des aktuellen Sterns von Suche 2. Der Endpunkt heißt „Ziel Suche 2“. Starte eine Newtonsuche in allen bisherigen Minima der Zielfunktion, nenne die Endpunkte Ziel_f^1, Ziel_f^2, ... Sortiere alle Elemente aus Ξ aufsteigend nach ihrem Zielfunktionswert. Starte im kleinsten eine Newtonsuche von Φ. Ist der Endpunkt weder „Ziel Suche 2“, noch einer der Ziel_f^1, Ziel_f^2, ... so ist x dieser Knoten. Falls nicht, so teste auf gleiche Weise den Knoten mit dem zweitkleinsten Zielfunktionswert, usw. Werden auf diese Art alle Elemente aus Ξ ausgeschlossen, so wähle den Knoten aus Ξ mit dem größten Minimalabstand zu den übrigen Knoten aus Ξ.

Analog dazu führen wir „$\tilde{x}$ =Finde Zentrum S_2“ durch, wenn das Zentrum des nächsten Sterns von Suche 2 ermittelt werden soll.

Weiterhin durchgeführt werden n_Φ lokale Suchen in den besten lokalen Minima des aktuellen Interpolanden Φ.

„$\{x_\Phi^1, \ldots, x_\Phi^{n_\Phi}\}$ =Lokale Suche $\Phi(n_{loc})$“: Ermittle alle lokalen Minima des Inter-

polanden Φ. Ordne diese aufsteigend nach dem jeweiligen Wert von Φ. Eliminiere alle, die in das selbe Ziel wie Suche 1 oder Suche 2 oder in ein bereits vorhandenes Minimum führen. Gehe dazu vor wie in „Finde Zentrum “. Wähle von den n_Φ übriggebliebenen die ersten n_{loc} aus. Gilt $n_\Phi < n_{loc}$, übriggeblieben, so werden entsprechend mehr Punkte für Algorithmus 8 zur Verfügung gestellt.

Es wurde bisher mehrfach erwähnt, daß lokale Minima der Zielfunktion nicht mehrfach angesteuert werden sollen. Doch wann geht man davon aus, daß ein lokales Minimun vorhanden ist?

„x_{min} =Erkenne Minimum (x)“: Wurden aus einem Differenzenstern mit Zentrum x Suchrichtungen ermittelt und dort ausgewertet, dann können wir

$$x_{min} := \text{argmin}\{f(S(x, h_g)), f(S(x, h_N)), f(S(x, h_H,))\}$$

bestimmen. Liegt dieser Punkt näher als 10^{-3} an x, so gilt er als lokales Minimum. Wenn nicht, wird $x_{min} := \varnothing$ ausgegeben.

Insgesamt entsteht so das Gesamtverfahren *GloPa*. Grob gesagt arbeitet das Verfahren in jeder Iteration vier Schritte ab:

1) Lokale Suche in den besten Stellen des Interpolanden, also Bestimmung der $x_\Phi^1, \ldots, x_\Phi^{n_\Phi}$ nach „Lokale Suche Φ“.

2) Bestimmung der globalen Suchpunkte $x_1, \ldots, x_{m_g}$ nach Algorithmus 8.

3) Lokale Suche dritter Ordnung:
Entweder „x =Finde Zentrum S_1“, „Bestimme $D_1(x)$“, „Bestimme $r_2(\tilde{x})$“, oder „$\tilde{x}$ =Finde Zentrum S_2“, „Bestimme $D_2(\tilde{x})$“, „Bestimme $r_1(x)$“.

4) Hochparallele Auswertung in $x_\Phi^1, \ldots, x_\Phi^{n_\Phi}$, $x_1, \ldots, x_{m_g}$ sowie dem Stern und den Suchrichtungen D bzw. S_m.

Wir gehen nun davon aus, daß n_{glob} Auswertungen in die globale Suche und n_{loc} Auswertungen in die lokale Suche durch Minimierung des Interpolanden investiert werden soll. Die Anzahl der Auswertungen für die Differenzensterne und Suchrichtungen bleibt etwa gleich. Wir erhalten insgesamt Algorithmus 9, in dem x_{min} die Menge der lokalen Minima der Zielfunktion bezeichnet, die wie oben beschrieben ermittelt wird. Wir testen dieses Verfahren am selben Testsatz wie GloSe. Auch das Abbruchkriterium ist gleich. In jedem Fall wird das globale Minimum gefunden, auch wenn teilweise erst in den falschen lokalen Minima gesucht wurde. Angegeben wird der Index i aus GloPa, als das Abbruchkriterium erfüllt war.

Algorithmus 9 GloPa

Input: Ξ, Y, f, n_{glob}, n_{loc}

Output: x^*

1: $i := 0$
2: $\Xi_0 := \Xi$, $Y_0 := Y$
3: $x_{min} = \varnothing$
4: **while** Abbruchbedingung nicht erfüllt **do**
5: Bestimme Interpolanden Φ der Form (4.2) und (4.3) für Ξ_i und Y_i
6: **if** i=0 **then**
7: $z_{D_1} :=$ argmin(Y)
8: Bestimme $D_1(z_{D_1})$
9: $x_\Phi^1, \ldots, x_\Phi^{n_\Phi}$ =Lokale Suche $\Phi(n_{loc})$
10: $x_{loc} := \{x_\Phi^1, \ldots, x_\Phi^{n_\Phi}\} \cup D_1(z_{D_1})$
11: **end if**
12: **if** $i \equiv 1$ mod 2 **then**
13: z_{D_2} :=Finde Zentrum D_2
14: Bestimme $D_2(z_{D_2})$
15: Bestimme $r_1(z_{D1})$
16: $x_\Phi^1, \ldots, x_\Phi^{n_\Phi}$ =Lokale Suche $\Phi(n_{loc})$
17: $x_{loc} := \{x_\Phi^1, \ldots, x_\Phi^{n_\Phi}\} \cup D_2(z_{D_2}) \cup r_1(z_{D1})$
18: **else**
19: z_{D_1} :=Finde Zentrum D_1
20: Bestimme $D_1(z_{D_1})$
21: Bestimme $r_2(z_{D2})$
22: $x_\Phi^1, \ldots, x_\Phi^{n_\Phi}$ =Lokale Suche $\Phi(n_{loc})$
23: $x_{loc} := \{x_\Phi^1, \ldots, x_\Phi^{n_\Phi}\} \cup D_1(z_{D_1}) \cup r_2(z_{D2})$
24: **end if**
25: $m_g := n_{glob} + (n_{loc} - n_\Phi)$
26: $x_1, \ldots, x_{m_g}$=Algorithmus 8(Ξ, Y, m_g)
27: $x_{glob} := x_1, \ldots, x_{m_g}$
28: $\Xi_{i+1} = \Xi_i \cup x_{loc} \cup x_{glob}$, $Y_{i+1} := Y_i \cup f(x_{loc}) \cup f(x_{glob})$
29: **if** $i \equiv 1$ mod 2 **then**
30: min_{neu} :=Erkenne Minimum (z_{D1})
31: **else**
32: min_{neu} :=Erkenne Minimum (z_{D2})
33: **end if**
34: $x_{min} := x_{min} \cup min_{neu}$
35: i:=i+1
36: $x^* :=$ argmin$\{f(x) \mid x \in \Xi_i\}$
37: **end while**

Dies ist gerade die Anzahl an parallelen Auswertungen.
Für ein 10-dimensionales Problem, wie in der ursprünglichen Problemstellung, wurde der Algorithmus nicht getestet. Ein Durchlauf einer Schleife in GloPa dauert dann etwa einen Tag. Es gibt aber dennoch Reaktoren mit $n \ll 10$ aber dennoch 1000 Waben oder sogar mehr und man erkennt die Lauffähigkeit der Methode schon an niederigerdimensionalen Problemen. Um aber eine zum Zehndimensionalen vergleichbare Situation zu schaffen, wurde folgende Abwandlung der Methode getestet: Zuerst wird angenommen, daß 2^n Punkte die Menge $[0,1]^n$ ungefähr so gut auffüllen wie 1000 Punkte den zehndimensionalen Einheitswürfel. Also haben wir in jedem Schritt 2^n Punkte zur Verfügung. Die Hälfte davon soll für das globale Abtasten verwendet werden. Etwa 40% kosten der Differenzenstern und die Suchrichtungen. Dies wurde dadurch simuliert, daß ca 10% der Auswertungen für die Lokale Suche durch den Interpolanden zur Verfügung gestellt werden. Die Halleypunkte x_{s_i} sowie die Auswertungen des Sterns werden nicht zu Ξ hinzugenommen, sondern nur der beste Punkt aus x_{s_i}. Es wurde dann je nach Dimension des Problems mit folgenden Größen gerechnet:

n	2^n	n_{glob}	$\#x_\Phi$
2	4	2	0
3	8	4	≤ 1
4	16	8	≤ 2
6	64	32	≤ 7

Es ist richtig, daß hier für $\#x_\Phi$ jeweils etwas mehr als 10% veranschlagt wurden. Dies wurde getan, damit diese lokale Suche überhaupt durchgeführt wird.
Es wurden folgende Ergebnisse erreicht:

Testfunktion	GloPa
Branin	6
Goldstein-Price	8
Hartmann3	7
Hartmann6	9
Shekel5	5
Shekel7	5
Shekel10	4

Natürlich ist die Anzahl der benötigten Schritte viel geringer als für das sequentielle Verfahren. Die Gesamtzahl der Auswertungen ist aber jeweils höher. Es ist trotzdem zu vermuten, daß man die Anzahl der benötigten Iterationen nicht

deutlich senken kann, wenn man nicht einen lokalen Sucher entwickelt, der signifikant besser ist als das Halley-Verfahren. Dies scheint ausgeschlossen, weil schon $\mathcal{V}$ kaum eine Verbesserung brachte. Außerdem müßte man schon apriori sehr gut wissen, wo das globale Minimum liegt, weil man neben $\mathcal{V}$ im 10 Dimensionalen entweder keine zweite Suche oder aber keine globale Suche laufen lassen kann.

4.2.3 Beweis der Globalen Konvergenz

Die beiden entwickelten Verfahren scheinen gute numerische Ergebnisse zu liefertn. Die Sicherheit, daß man das globale Minimum auch wirklich immer findet, wäre dennoch wünschenswert. Sie kann zumindest im Eindimensionalen gezeigt werden. Es kann weiterhin gezeigt werden, daß auch im Mehrdimensionalen ein vergleichbares Resultat sehr plausibel ist.
In aller Regel bauen Konvergenzbeweise in der globalen Optimierung auf dem folgenden Satz auf, der in [38] angegeben und dort auch bewiesen wird.

Satz 4.1. *Sei $\mathcal{D}$ eine kompakte Teilmenge des $\mathbb{R}^n$ und $\{x_i\}_{i\geq 1}$ die von einem Algorithmus $\mathcal{A}$ erzeugte Folge, deren Elemente alle in $\mathcal{D}$ liegen. Es gilt: Ist die von $\mathcal{A}$ generierte Folge dicht in $\mathcal{D}$, so konvergiert sie gegen das globale Minimum jeder auf $\mathcal{D}$ stetigen Funktion f.*

Der Satz gibt keine Auskunft darüber, wie schnell ein Verfahren das lokale Minimum findet. Man könnte ihn eher als Indiz für eine gewisse Ineffizienz werten, da keine Konvergenz innerhalb einer vorgegebenen Laufzeit prognostiziert werden kann. Er sagt lediglich aus, daß der ganze Suchraum dicht ausgeschöpft wird und man so zwangläufig in der Nähe des globalen Minimums auswerten muß. Dennoch wird er oft verwendet.
Auch im vorliegenden Fall wird von diesem Ergebnis Gebrauch gemacht. Ist die globale Konvergenz von GloSe gezeigt, so folgt automatisch auch die von GloPa, weil der erste globale Punkt von GloPa gleich dem von GloSe ist. Man erreicht im Eindimensionalen folgendes Resultat:

Satz 4.2. *Sei $[a,b]$ ein abgeschlossenes Intervall in $\mathbb{R}$. Sei f auf einer offenen Obermenge von $[a,b]$ differenzierbar und f' genüge auf $[a,b]$ einer Lipschitz-Bedingung. Dann konvergiert die von GloSe erzeugte Folge gegen das globale Minimum von f auf $[a,b]$.*

Beweis: Wir klären noch einmal genau den Begriff der Dichtheit: Eine Folge ist genau dann dicht in $[a,b]$, wenn es für alle $x \in [a,b]$ und alle $\varepsilon > 0$ ein Element

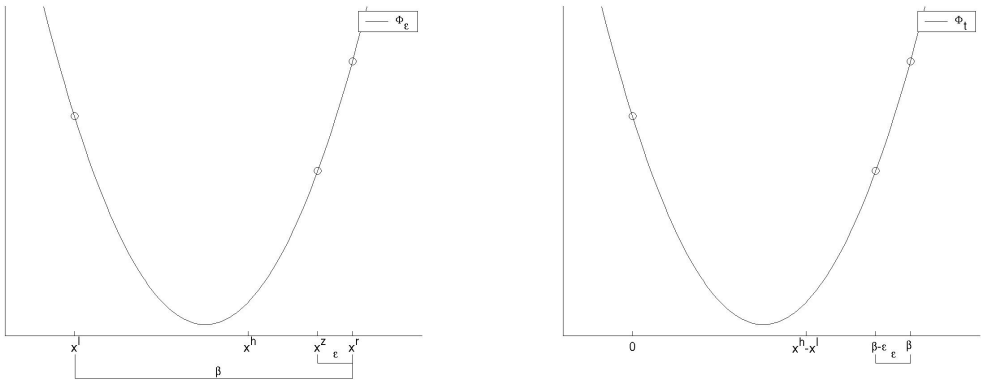


Abbildung 4.3: Natürlicher Kubischer Spline in Häufungspunktnähe und transformierter Spline

der Folge gibt, dessen Entfernung zu x echt kleiner als ε ist. Ist eine Folge nicht dicht, dann gibt es also ein offenes Intervall, das komplett in $[a, b]$ liegt, und kein Element der Folge enthält.
Um die Konvergenz zu zeigen, führen wir einen Widerspruchsbeweis. Es sei also die von GloSe erzeugte Folge $\{x_i\}_{i\geq 1}$ nicht dicht in $[a, b]$, womit ein $]\tilde{x} - \delta, \tilde{x} + \delta[$ existiert, das kein Element dieser Folge enthält.
Wir betrachten nun die Teilfolge von Elementen, die durch die Minimierung der Krümmung ausgesucht werden und nennen diese $\{x_i^h\}_{i\geq 1}$. Dadurch liegt eine Folge in einer kompakten Menge vor. Nach dem Satz von Bolzano und Weierstraß hat diese Folge einen Häufungspunkt x^h.
Wir betrachten nun folgende Situation, die irgendwann im Laufe des Algorithmus auftreten muß: Bisher hat der Algorithmus $m + 1$ Auswertungsstellen ermittelt. In x^h selber wurde noch nicht ausgewertet (Der Fall, daß in x^h schon ausgewertet wurde, wird anschließend behandelt) und es gibt einen ersten Knoten x^r rechts von x^h und einen ersten x^l links von x^h. Der Abstand der beiden sei β, und das von den beiden begrenzte das Intervall heiße J. Dadurch, daß x^h ein Häufungspunkt der Folge $\{x_i^h\}_{i\geq 1}$ ist, wird das Verfahren als interessantesten Punkt irgendwann einen aus dem Inneren von J auswählen, den wir x^z nennen. Der Abstand von x^z vom rechten Intervallende von J sei ε. Vom Verfahren wurde also der interpolierende natürliche kubische Spline durch die bisherigen Knoten und zusätzlich noch durch (x^z, y^z) gelegt. Diesen Interpolanden nennen wir ab sofort $\Phi_\varepsilon(x)$. Abbildung 4.3 zeigt die Situation.
Weil der natürliche kubische Spline im Eindimensionalen eine abschnittweise Darstellung besitzt, kann man eine untere Schranke für

$$\int_a^b \left(\Phi_\varepsilon''(x)\right)^2 dx,$$

und damit für die Interessantheit plus der Barriere angeben. Dazu werden die bisherigen durch den Algorithmus ermittelten Knoten um x^z ergänzt, anschließend aufsteigend sortiert und dann als x_i mit $i = 0, \ldots, m+1$ bezeichnet. Diese werden nun einfach um x_l so verschoben, daß x^l auf die Null transformiert wird. x^z wird demnach auf $\beta - \varepsilon$ verschoben und x^r auf β. Formal definieren wir

$$x_i^t := x_i - x^l.$$

Natürlich sollen die y-Werte dabei nicht verändert werden. Der auf den x_i^t interpolierende natürliche kubische Spline ist also der um x^l transformierte Φ_ε, den wir Φ_t nennen. Abbildung 4.3 erleichtert die Vorstellung. Es gilt dann natürlich

$$\int_{a-x^l}^{b-x^l} \left(\Phi_t''(x)\right)^2 dx = \int_a^b \left(\Phi_\varepsilon''(x)\right)^2 dx,$$

weshalb wir ab sofort nur noch das Integral über $(\Phi_t''(x))^2$ untersuchen. Wegen $(\Phi_t''(x))^2 \geq 0$ auf ganz $[a - x^l, b - x^l]$ kann man die Betrachtung durch

$$\int_{a-x^l}^{b-x^l} \left(\Phi_t''(x)\right)^2 dx \geq \int_0^\beta \left(\Phi_t''(x)\right)^2 dx = \int_0^{\beta-\varepsilon} \left(\Phi_t''(x)\right)^2 dx + \int_{\beta-\varepsilon}^\beta \left(\Phi_t''(x)\right)^2 dx$$

vereinfachen.
Um die Integrale zwischen $[0, \beta - \varepsilon]$ und $[\beta - \varepsilon, \beta]$ explizit angeben zu können, müssen wir noch die Darstellung von Φ_t klären. Wir gehen dazu ähnlich vor wie in [37], und bauen den Spline Φ_t folgendermaßen auf: Auf $[a - x^t, b - x^t]$ soll er zweimal stetig differenzierbar sein. Da er abschnittsweise (also jeweils zwischen zwei aufeinanderfolgenden Knoten x_i^t und x_{i+1}^t) ein Polynom dritten Grades ist und $\Phi_t''(x_i^t) = M_i$ für noch zu bestimmendes M_i gelten soll, ergibt sich die Darstellung

$$\Phi_t''(x) = \frac{x - x_i^t}{x_{i+1}^t - x_i^t} M_{i+1} + \frac{x_{i+1}^t - x}{x_{i+1}^t - x_i^t} M_i \quad \text{für} \quad x \in [x_i^t, x_{i+1}^t].$$

Weil nach unserer Definition $x_i^t = 0$ für irgendein $0 \leq i \leq m+1$, definieren wir der Übersichtlichkeit wegen dann $M_0^t := M_i, M_1^t := M_{i+1}$ und $M_2^t := M_{i+2}$. Damit gelten

$$\Phi_t''(x) = \frac{x}{\beta-\epsilon}M_1^t + \frac{\beta-\varepsilon-x}{\beta-\epsilon}M_0^t \quad \text{für} \quad x \in [0, \beta-\epsilon],$$

bzw.

$$\Phi_t''(x) = \frac{x-\beta+\epsilon}{\epsilon}M_2^t + \frac{\beta-x}{\epsilon}M_1^t \quad \text{für} \quad x \in [\beta-\epsilon, \beta].$$

Weil $\Phi_t''(x)$ auf $[0, \beta-\epsilon]$ und $[\beta-\epsilon, \beta]$ jeweils ein Polynom vom Grad zwei in x ist, kann man leicht die Stammfunktionen bestimmen. Etliche leicht nachvollziehbare Umformungen führen zu

$$\begin{aligned}
&\int_0^{\beta-\epsilon} \left(\Phi_t''(x)\right)^2 dx + \int_{\beta-\epsilon}^{\beta} \left(\Phi_t(x)\right)^2 dx \\
&= \frac{\left(x(M_1^t - M_0^t) + M_0^t(\beta-\varepsilon)\right)^3}{3(M_1^t - M_0^t)\left(\varepsilon-\beta\right)^2}\Bigg|_0^{\beta-\varepsilon} + \frac{\left(x(M_1^t - M_2^t) + M_2^t(\beta-\varepsilon) - \beta M_1^t\right)^3}{3(\varepsilon^2(M_1^t - M_2^t))}\Bigg|_{\beta-\varepsilon}^{\beta} \\
&= \frac{1}{3}\left((\beta-\varepsilon)((M_0^t)^2 + M_0^t M_1^t) + \beta(M_1^t)^2 + \epsilon(M_1^t M_2^t + (M_2^t)^2)\right) \qquad (4.1)
\end{aligned}$$

Dies scheint zunächst nicht hilfreich zu sein, denn es liegt eine quadratische Form in (M_0^t, M_1^t, M_2^t) ohne linearen Teil und ohne affine Verschiebung vor, die einer singulären Matrix entspringt. Damit könnte man unten höchstens durch null abschätzen, was ohnehin schon bekannt war und auch nicht verwundert, weil noch gar keine Funktionswerte in die Darstellung eingegangen sind.
Man kann unter Berücksichtigung der y-Werte aber M_0^t, M_1^t und M_2^t in Beziehung zueinander setzen: Dadurch, daß $\Phi_t(x)$ auf $[a-x^t, b-x^t]$ zweimal stetig differenzierbar sein und darüber hinaus interpolieren soll, stehen die Momente in folgender Beziehung zueinander (Siehe [37], wie dort setzen wir $h_{i+1} = x_{i+1}^t - x_i^t$):

$$M_i h_{i+1} + 2(h_{i+1} + h_{i+2})M_{i+1} + h_{i+2}M_{i+2} = 6\left(\frac{y_{i+2} - y_{i+1}}{h_{i+2}} - \frac{y_{i+1} - y_i}{h_{i+1}}\right). \quad (4.2)$$

Dieser Term ist unabhngig von den Randbedingungen, die erst später in die Konstruktion von Φ_t eingehen werden. Alles, was folgt, gilt also insbesondere auch für die Randbedingungen des natürlichen kubischen Splines. Für M_0^t, M_1^t und M_2^t bedeutet dies mit $\Delta^0 := y_i - y_{i+1}$ und $\Delta^1 := y_{i+2} - y_{i+1}$

$$M_0^t(\beta-\varepsilon) + 2\beta M_1^t + \varepsilon M_2^t = 6\left(\frac{\Delta^1}{\varepsilon} - \frac{-\Delta^0}{\beta-\varepsilon}\right),$$

was nach M_0^t aufgelöst wird. Man beachte an dieser Stelle, daß sowohl Δ^0 als auch Δ^1 so gewählt wurden, daß sie echt größer null sind.

$$M_0^t = \frac{6}{\beta - \varepsilon}\left(\left(\frac{\Delta^1}{\varepsilon} - \frac{-\Delta^0}{\beta - \varepsilon}\right) - 2\beta M_1^t - \varepsilon M_2^t\right)$$

Setzt man dies in Gleichung (4.1) ein, so erhält man nach einiger Rechnung

$$\begin{aligned}
I_t \;=\; & (M_1^t)^2 \underbrace{\frac{-111\beta^2\varepsilon^4 - 11\beta\varepsilon^5 + 255\beta^3\varepsilon^3 - 133\varepsilon^2\beta^4}{3(\varepsilon^2(\beta-\varepsilon)^3)}}_{=:a_{1,1}} \\
& + M_1^t M_2^t \underbrace{\frac{273\beta^2\varepsilon^4 - 139\beta^3\varepsilon^3 - 129\beta\varepsilon^5 - 5\varepsilon^6}{3(\varepsilon^2(\beta-\varepsilon)^3)}}_{=:2a_{2,1} \text{ und } =:2a_{1,2}} \\
& + (M_2^t)^2 \underbrace{\frac{-33\beta^2\varepsilon^4 - 35\varepsilon^6 + 69\beta\varepsilon^5 - \beta^3\varepsilon^3}{3(\varepsilon^2(\beta-\varepsilon)^3)}}_{=:a_{2,2}} \\
& + M_1^t \underbrace{\frac{-2(-21\delta\beta\varepsilon^3 + 22\alpha\beta\varepsilon^3 - 23\alpha\beta^2\varepsilon^2 + 45\delta\beta^2\varepsilon^2 - 23\delta\beta^3\varepsilon - \delta\varepsilon^4 + \alpha\varepsilon^4)}{\varepsilon^2(\beta-\varepsilon)^3}}_{=:b_1} \\
& + M_2^t \underbrace{\frac{-12(-2\delta\varepsilon^4 + 2\alpha\varepsilon^4 - 2\delta\beta^2\varepsilon^2 + 4\delta\beta\varepsilon^3 - 2\alpha\beta\epsilon^3)}{\varepsilon^2(\beta-\varepsilon)^3}}_{=:b_2} \\
& \underbrace{-12\frac{\delta^2\beta^2 - 2\delta^2\beta\varepsilon + 2\alpha\beta\delta\varepsilon + \delta^2\varepsilon^2 - 2\alpha\varepsilon^2\delta + \alpha^2\varepsilon^2}{\varepsilon^2(-\beta+\varepsilon)^3}}_{=:c_t}
\end{aligned}$$

was in dieser Darstellung noch sehr unübersichtlich ist. Es wird durch die Klammern aber schon die Quadratische Form, als die man dies schreiben kann, angedeutet. Übernimmt man die Definitionen aus den Klammern, so gilt dann mit

$$A_t := \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}, \quad b_t := \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad \text{und} \quad c_t \text{ wie oben}$$

die Gleichung

$$I_t = (M_1^t, M_2^t) A_t (M_1^t, M_2^t)^T + (M_1^t, M_2^t) b_t + c_t. \tag{4.3}$$

Aufgrund seiner Symmetrie hat A_t zwei reelle Eigenwerte λ_1^t und λ_2^t, die man explizit angeben kann:

$$\begin{aligned}\lambda_{1,2}^t &= \frac{420\varepsilon^2 + 144\beta\varepsilon + 1596\beta^2}{72(\beta-\varepsilon)} \\ &\quad \pm \frac{12\sqrt{1250\varepsilon^4 + 690\beta\epsilon^3 + 10111\beta^2\varepsilon^2 + 2660\beta^3\varepsilon + 17689\beta^4}}{72(\beta-\varepsilon)},\end{aligned}$$

wobei λ_1^t den Eigenwert meint, bei dem die Wurzel addiert wird. Wegen $\beta > \varepsilon$ ist dann natürlich $\lambda_1^t > 0$ für $\varepsilon > 0$. Zur Untersuchung von λ_2^t nutzen wir wieder $\beta > \varepsilon$: Es gilt dann natürlich $\beta = \varepsilon + \gamma$ mit $\gamma > 0$. Setzen wir die ein, so erhalten wir

$$\begin{aligned}\gamma_2^t &= \frac{180\varepsilon^2 + 278\varepsilon\gamma + 133\gamma^2}{6\gamma} \\ &\quad - \frac{\sqrt{32400\varepsilon^4 + 99648\varepsilon^3\gamma + 124225\varepsilon^2\gamma^2 + 73416\varepsilon\gamma^3 + 17689\gamma^4}}{6\gamma}\end{aligned}$$

Nun schreiben wir $180\varepsilon^2 + 278\varepsilon\gamma + 133\gamma^2$ etwas um:

$$180\varepsilon^2 + 278\varepsilon\gamma + 133\gamma^2 = \sqrt{\left(180\varepsilon^2 + 278\varepsilon\gamma + 133\gamma^2\right)^2}$$

$$= \sqrt{32400\varepsilon^4 + 100080\varepsilon^3\gamma + 125164\varepsilon^2\gamma^2 + 73948\varepsilon\gamma^3 + 17689\gamma^4}$$

Weil in dieser Wurzel nur positive Summanden auftreten und der Vorfaktor eines jeden Summanden größer gleich als der Vorfaktor des entsprechenden Summanden in der Wurzel mit negativem Vorzeichen ist, muß λ_2^t echt größer null sein. Damit ist A_t positiv definit und wir können (4.3) nach unten abschätzen: Diese Quadratische Form hat das einzige globale Minimum in

$$\begin{pmatrix} M_1^{min} \\ M_2^{min} \end{pmatrix} = -\frac{A_t^{-1} b_t}{2} = \begin{pmatrix} \frac{12(-5\delta\varepsilon^2+5\alpha\varepsilon^2-23\alpha\beta\varepsilon+28\delta\beta\varepsilon-23\delta\beta^2)}{(\varepsilon(-\beta+\varepsilon)(25\varepsilon^2-125\beta\varepsilon+532\beta^2)} \\ \frac{-30(\delta\beta-\delta\varepsilon+\alpha\varepsilon)}{(\varepsilon(25\varepsilon^2-125\beta\varepsilon+532\beta^2))} \end{pmatrix}.$$

Einsetzen in (4.3) liefert

$$I_t(M_1^{min}, M_2^{min}) = 36\frac{\beta(\delta\beta - \delta\varepsilon + \alpha\varepsilon)^2}{\varepsilon^2(25\varepsilon^2 - 125\beta\varepsilon + 532\beta^2)(\beta-\varepsilon)^2}.$$

Wir zerlegen diesen Ausdruck in $36 \cdot I_t^1 \cdot I_t^2$, wobei

$$I_t^1 := \frac{\beta}{25\varepsilon^2 - 125\beta\varepsilon + 532\beta^2} \quad \text{bzw.} \quad I_t^2 := \frac{(\delta\beta - \delta\varepsilon + \alpha\varepsilon)^2}{\varepsilon^2(\beta - \varepsilon)^2}.$$

Zunächst schätzen wir I_t^1 nach oben ab. Mit quadratischer Ergänzung erhalten wir:

$$I_t^1 = \frac{\beta}{\left(5\varepsilon - \frac{25}{2}\beta\right)^2 + \frac{1503}{4}\beta^2} \overset{\beta>\varepsilon}{\geq} \frac{\beta}{\left(\frac{25}{2}\beta\right)^2 + \frac{1503}{4}\beta^2} = \frac{4}{2128\beta}$$

I_t^1 kann also nur klein werden, wenn β groß wird. β ist aber durch $b - a$ nach oben beschränkt. Deshalb können wir $I_t^1 \geq C_1 > 0$ für eine nur von der Intervallgröße abhängigen Konstante C_1 schreiben. Insbesondere ändert sich C_1 nicht im Verfahrensverlauf.
Für I_t^2 gilt wegen $\beta > \varepsilon$ und $\alpha, \delta, \varepsilon > 0$

$$I_t^2 = \frac{(\delta(\beta - \varepsilon) + \alpha\varepsilon)^2}{\varepsilon^2(\beta - \varepsilon)^2} \geq \frac{(\delta(\beta - \varepsilon))^2}{\varepsilon^2(\beta - \varepsilon)^2} = \frac{\delta^2}{\varepsilon^2}.$$

I_t^2 kann also nur klein werden für kleine δ. δ ist aber wegen (4.1) nach unten beschränkt. Also gilt auch $I_t^2 \geq \frac{C_2}{\varepsilon^2}$ mit einem C_2, das sich nicht mit dem Verfahrensverlauf ändert.
Dadurch, daß x^h ein Häufungspunkt ist, tritt die beschriebene Situation für beliebig kleine $\varepsilon > 0$ auf, I_t wächst also im Verfahrensverlauf über alle Grenzen.
Wir hatten zu Beginn des Beweises den Fall, daß in x^h schon ausgewertet wurde, zurückgestellt. Rückblickend war dies keine Beeinträchtigung der Allgemeinheit. Wurde bereits in x^h ausgewertet, so nimmt x^h die Rolle von x^l bzw. x^r ein und der Beweis funktioniert ganz genau so.
Es wurde zu Beginn weiterhin angenommen, daß es ein Intervall $]\tilde{x} - \delta, \tilde{x} + \delta[$ gibt, das kein Element aus $\{x_i^h\}$ enthält. Wir zeigen nun, daß dies nicht sein kann.
Dazu kehren wir wieder zu den am Beginn des Beweises erwähnten $m+1$ Knoten zurück, die bisher vom Verfahren ermittelt wurden. Diese werden diesmal nicht um x^z, sondern um $\tilde{x}$ samt y^z ergänzt, anschließend aufsteigend sortiert und diesmal als $\tilde{x}_i$ mit $i = 0, \ldots, m+1$ bezeichnet. Wir prüfen nun, für wie interessant das Verfahren den Punkt $\tilde{x}$ hält, wofür der interpolierende kubische Spline $\tilde{\Phi}$ durch die $(\tilde{x}_i, y_i)$ gelegt wird.
Wie im ersten Teil des Beweises wird erneut die Darstellung

$$\tilde{\Phi}''(x) = \frac{x - \tilde{x}_i}{\tilde{x}_{i+1} - \tilde{x}_i}\tilde{M}_{i+1} + \frac{\tilde{x}_{i+1} - x}{\tilde{x}_{i+1} - \tilde{x}_i}\tilde{M}_i \quad \text{für} \quad x \in [\tilde{x}_i, \tilde{x}_{i+1}].$$

gewählt. Man erkennt dann

$$\begin{aligned}
\int_a^b \left(\tilde{\Phi}''(x)\right)^2 dx &= \sum_{i=0}^{m} \int_{\tilde{x}_i}^{\tilde{x}_{i+1}} \tilde{\Phi}''(x)dx \\
&= \sum_{i=0}^{m} \int_{\tilde{x}_i}^{\tilde{x}_{i+1}} \left(\frac{x - \tilde{x}_i}{\tilde{x}_{i+1} - \tilde{x}_i}\tilde{M}_{i+1} + \frac{\tilde{x}_{i+1} - x}{\tilde{x}_{i+1} - \tilde{x}_i}\tilde{M}_i\right)^2 dx \\
&\vdots \\
&= \sum_{i=0}^{m} \frac{1}{3}(\tilde{x}_{i+1} - \tilde{x}_i)\left(\tilde{M}_i^2 + \tilde{M}_i\tilde{M}_{i+1} + \tilde{M}_{i+1}^2\right) \\
&\leq \sum_{i=0}^{m} (\tilde{x}_{i+1} - \tilde{x}_i)\tilde{M}_{max}^2 \\
&= (b-a)\tilde{M}_{max}^2,
\end{aligned} \tag{4.4}$$

wobei $\tilde{M}_{max}$ der Betrag des betragsgrößten $\tilde{M}_i$ ist. Wir haben somit zu untersuchen, ob und wie sich $\tilde{M}_{max}$ nach oben abschätzen läßt. Mit diesem Ziel betrachten wir das Lineare Gleichungssystem, das im Fall des natürlichen kubischen Splines die Momente definiert und legen zunächst $\tilde{h}_{i+1} = \tilde{x}_{i+1} - \tilde{x}_i$ fest. Durch die Einsortierung von $\tilde{x}$ haben wir einen Index k mit $\tilde{x}_k = \tilde{x}$, damit folgen automatisch $y_k = y^z$, $\tilde{h}_k = \tilde{h}_{k+1} = \delta$. Die Momente hängen wieder miteinander zusammen wie in (4.2), nur daß wir diesmal jede Zeile mit

$$\frac{1}{\tilde{h_1} + \tilde{h_2}}$$

durchmultiplizieren. Das enstehende Gleichungssystem lautet dann $\tilde{A}\tilde{M} = \tilde{b}$, die Struktur seiner Bestandteile ist folgendermaßen.

$$\tilde{A} = \begin{pmatrix} 2 & \frac{\tilde{h}_2}{\tilde{h}_1+\tilde{h}_2} & & & & & & & \\ \frac{\tilde{h}_2}{\tilde{h}_2+\tilde{h}_3} & \ddots & \ddots & & & & & & \\ & \ddots & \ddots & \ddots & & & & & \\ & & \frac{\tilde{h}_{k-1}}{\tilde{h}_{k-1}+\delta} & 2 & \frac{\delta}{\tilde{h}_{k-1}+\delta} & & & & \\ & & & \frac{1}{2} & 2 & \frac{1}{2} & & & \\ & & & & \frac{\delta}{\delta+\tilde{h}_{k+2}} & 2 & \frac{\tilde{h}_{k+2}}{\delta+\tilde{h}_{k+2}} & & \\ & & & & & \ddots & \ddots & \ddots & \\ & & & & & & \ddots & \ddots & \frac{\tilde{h}_m}{\tilde{h}_{m-1}+\tilde{h}_m} \\ & & & & & & & \frac{\tilde{h}_m}{\tilde{h}_m+\tilde{h}_{m+1}} & 2 \end{pmatrix},$$

$$\tilde{M} = \begin{pmatrix} \tilde{M}_1 \\ \vdots \\ \tilde{M}_{k-2} \\ \vdots \\ \tilde{M}_{k+2} \\ \vdots \\ \tilde{M}_m \end{pmatrix} \quad \text{und} \quad \tilde{b} = 6 \begin{pmatrix} \frac{y_2-y_1}{(\tilde{h}_1+\tilde{h}_2)\tilde{h}_2} - \frac{y_1-y_0}{(\tilde{h}_1+\tilde{h}_2)\tilde{h}_1} \\ \vdots \\ \frac{y^z-y_{k-1}}{(\tilde{h}_{k-1}+\delta)\delta} - \frac{y_{k-1}-y_{k-2}}{(\tilde{h}_{k-1}+\delta)\tilde{h}_{k-1}} \\ \frac{y_{k+1}-y^z}{2\delta^2} - \frac{y^z-y_{k-1}}{2\delta^2} \\ \frac{y_{k+2}-y_{k+1}}{(\delta+\tilde{h}_{k+2})\tilde{h}_{k+2}} - \frac{y_{k+1}-y^z}{(\delta+\tilde{h}_{k+2})\delta} \\ \vdots \\ \frac{y_{m+1}-y_m}{(\tilde{h}_m+\tilde{h}_{m+1})h_{\tilde{m}+1}} - \frac{y_M-y_{m-1}}{(\tilde{h}_m+\tilde{h}_{m+1})\tilde{h}_m} \end{pmatrix}.$$

Dieses Lineare Gleichungssystemn hat eine für uns sehr interessante Eigenschaft: Sei $\tilde{b}_{max}$ der Betrag des betragsgrößten Eintrags in $\tilde{b}$. Dann gilt

$$\tilde{A}\tilde{M} = \tilde{b} \quad \Rightarrow \quad \tilde{M}_{max} \leq \tilde{b}_{max}.$$

Zum Nachweis wählen wir dazu den Index r (Das hat nichts mit y^r zu tun und gilt nur für diesen Teilbeweis) so, daß $|M_r| = \tilde{M}_{max}$. Wegen $\tilde{A}\tilde{M} = \tilde{b}$ haben wir

$$\begin{aligned} & \tilde{M}_r \frac{\tilde{h}_{r+1}}{\tilde{h}_{r+1}+\tilde{h}_{r+2}} + 2\tilde{M}_{r+1} + \frac{\tilde{h}_{r+2}}{\tilde{h}_{r+1}+\tilde{h}_{r+2}} \tilde{M}_{r+2} \\ = \quad & 6\left(\frac{y_{r+2}-y_{r+1}}{(\tilde{h}_{r+1}+\tilde{h}_{r+2})\tilde{h}_{r+2}} - \frac{y_{r+1}-y_r}{(\tilde{h}_{r+1}+\tilde{h}_{r+2})\tilde{h}_{r+1}} \right) = \tilde{b}_r \end{aligned}$$

Mit

$$\frac{\tilde{h}_{r+1}}{\tilde{h}_{r+1}+\tilde{h}_{r+2}} + \frac{\tilde{h}_{r+2}}{\tilde{h}_{r+1}+\tilde{h}_{r+2}} = 1$$

sehen wir dann:

$$
\begin{aligned}
\tilde{M}_{max} &= |\tilde{M}_r| = 1 \cdot |\tilde{M}_r| \\
&= \left(2 - \frac{\tilde{h}_{r+1}}{\tilde{h}_{r+1}+\tilde{h}_{r+2}} - \frac{\tilde{h}_{r+2}}{\tilde{h}_{r+1}+\tilde{h}_{r+2}}\right)|\tilde{M}_r| \\
&= 2|\tilde{M}_r| - \frac{\tilde{h}_{r+1}}{\tilde{h}_{r+1}+\tilde{h}_{r+2}}|\tilde{M}_r| - \frac{\tilde{h}_{r+2}}{\tilde{h}_{r+1}+\tilde{h}_{r+2}}|\tilde{M}_r| \\
&\leq 2|\tilde{M}_r| - \frac{\tilde{h}_{r+1}}{\tilde{h}_{r+1}+\tilde{h}_{r+2}}|\tilde{M}_{r-1}| - \frac{\tilde{h}_{r+2}}{\tilde{h}_{r+1}+\tilde{h}_{r+2}}|\tilde{M}_{r+1}| \\
&\leq \left|2\tilde{M}_r + \frac{\tilde{h}_{r+1}}{\tilde{h}_{r+1}+\tilde{h}_{r+2}}\tilde{M}_{r-1} + \frac{\tilde{h}_{r+2}}{\tilde{h}_{r+1}+\tilde{h}_{r+2}}\tilde{M}_{r+1}\right| \\
&= |\tilde{b}_r| \leq \tilde{b}_{max},
\end{aligned}
\tag{4.5}
$$

wobei die vorletzte Ungleichung aus der Dreiecksungleichung folgt. Dies hat unmittelbar zur Folge, daß man $\tilde{M}_{max}$ nach oben abschätzen kann, wenn $\tilde{b}_{max}$ beschränkt bleibt. Dies wiederum kann man zeigen, wenn man berücksichtigt, daß die Zielfunktion f stetig differenzierbar ist und die Ableitung einer Lipschitzbedingung genügt. Wir betrachten dazu ein beliebiges

$$
\tilde{b}_i = 6\left(\frac{y_{i+1}-y_i}{(\tilde{h}_i+\tilde{h}_{i+1})\tilde{h}_{i+1}} - \frac{y_i-y_{i-1}}{(\tilde{h}_i+\tilde{h}_{i+1})\tilde{h}_i}\right),
$$

in das y^z nicht eingeht. Wegen $f \in C^1([a,b])$ kann man den Mittelwertsatz anwenden. Es gibt also $\xi_i \in [x_{i-1}, x_i]$ und $\xi_{i+1} \in [x_i, x_{i+1}]$ mit

$$
f'(\xi_i) = \frac{y_i - y_{i-1}}{\tilde{h}_i} \quad \text{und} \quad f'(\xi_{i+1}) = \frac{y_{i+1}-y_i}{\tilde{h}_{i+1}}.
$$

Zusammen mit dem offensichtlichen $\xi_{i+1} - \xi_i \leq \tilde{h}_i + \tilde{h}_{i+1}$ und der für f' geltenden Lipschitzbedingung mit Konstante L_2 erhält man

$$
\tilde{b}_i = 6\left(\frac{f'(\xi_{i+1}) - f'(\xi_i)}{\tilde{h}_i + \tilde{h}_{i+1}}\right) \leq 6\left(\frac{f'(\xi_{i+1}) - f'(\xi_i)}{\xi_{i+1} - \xi_i}\right) \leq 6L_2
$$

Es fehlt noch die Betrachtung von $\tilde{b}_{k-1}, \tilde{b}_k$ und $\tilde{b}_{k+1}$, in die y^z eingeht. Dadurch, daß y^z nicht notwendigerweise gleich $f(\tilde{x})$ ist, müssen wir hier anders vorgehen: Es sind sechs Brüche zu untersuchen. Auf die beiden, in denen kein y^z vorkommt, wendet man wieder wegen des Mittelwertsatzes als und folgert

$$\frac{y_{k-1}-y_{k-2}}{(\tilde{k}_{k-1}+\delta)\tilde{k}_{k-1}} = \frac{f'(\xi_{k-1})}{\tilde{k}_{k-1}+\delta} < \frac{f'(\xi_{k-1})}{\delta} \leq C_1 \quad \text{bzw.} \quad \frac{y_{k-1}-y_{k-2}}{(\tilde{k}_{k-1}+\delta)\tilde{k}_{k-1}} < C_2,$$

wobei beide Konstanten nicht von der Verteilung der Knoten abhängen. Für die vier anderen Brüche gilt:
y^z bleibt wegen (4.1) im Verfahrensverlauf beschränkt, deswegen sind die auftretenden Zähler vom Betrag her alle nach oben beschränkt. Keiner der auftretenden Nenner kann kleiner als δ^2 werden. Also sind die vier Brüche vom Betrag durch eine Konstante C_3 und damit $\tilde{b}_{k-1}, \tilde{b}_k$ und $\tilde{b}_{k+1}$ durch ein C_4 nach oben beschränkt. Zusammen mit (4.5) und (4.6) gilt

$$\int_a^b \left(\tilde{\Phi}(x)\right)^2 dx < (b-a)\max\{36L_2^2, C_4^2\}. \tag{4.6}$$

Im Laufe der Iteration wird $\tilde{x}$ also nicht uninteressanter als eine bestimmte Grenze. Dies ändert sich auch durch die Barrierefunktion nicht, die ja in diesem Punkt im Verfahrensverlauf immer kleiner wird. Trozdem wird dort nicht ausgewertet. Im Gegensatz dazu steigt die Uninteressantheit der x^z in's Unendliche an, dort wird aber ausgewertet. Das Verfahren wertet aber immer im interessantesten Punkt aus, das ist ein Widerspruch. Also kann ein Intervall, in dem das Verfahren nie auswertet, nicht existieren. Damit ist die vom Verfahren erzeugte Folge dicht und der Algorithmus konvergiert global.

□

Anmerkung: Der Aufbau des Beweises läßt erahnen, warum man im Beliebigdimensionalen wahrscheinlich mehr Schwierigkeiten haben wird als eben: Dort kann die Darstellung des Interpolanden als abschnittsweises Polynom nicht verwendet werden. Aber auch die Darstellung als Summe Radialer Basisfunktionen erlaubt eine Betrachtung, die Hoffnung auf mehr macht.
Versucht man damit einen ähnlichen Beweis, so geht man von bisher $m-1$ vorhandenen Auswertungsstellen $\Xi = \{x_2, \ldots, x_m\}$ und dort gegebenen Auswertungen $Y = \{y_2, \ldots, y_m\}$ aus. Wir betrachten wieder einen Häufungspunkt x^h und einen um ε von x^l entfernten Punkt x^z. O.B.d.A. seien $x_2 = x_l$, $x_1 := x^z$ und $y_1 := y^z$. Erneut sei die Differenz von y^z und $f(x_l)$ nach unten beschränkt durch δ. Der zugehörige Interpoland heißt wieder Φ_ε. Dann lautet die Matrix des Linearen

Gleichungssystems (1.9)

$$\begin{pmatrix}
0 & \varepsilon^3 & \|x_z - x_3\|^3 & \cdots & \|x_z - x_m\|^3 & 1 & x_z^1 & \cdots & x_z^n \\
\varepsilon^3 & 0 & \|x_l - x_3\|^3 & \cdots & \|x_l - x_m\|^3 & 1 & x_l^1 & \cdots & x_l^n \\
\|x_3 - x_z\|^3 & \|x_3 - x_l\|^3 & 0 & \cdots & \|x_3 - x_m\|^3 & 1 & x_3^1 & \cdots & x_3^n \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & & \vdots \\
\|x_m - x_z\|^3 & \|x_m - x_l\|^3 & \|x_m - x_3\|^3 & \cdots & 0 & 1 & x_m^1 & \cdots & x_m^n \\
1 & 1 & 1 & \cdots & 1 & 0 & \cdots & \cdots & 0 \\
x_z^1 & x_l^1 & x_3^1 & \cdots & x_m^1 & \vdots & & & \vdots \\
\vdots & \vdots & \vdots & & \vdots & \vdots & & & \vdots \\
x_z^n & x_l^n & x_3^n & \cdots & x_m^n & 0 & \cdots & \cdots & 0
\end{pmatrix}$$

und Lösungsvektor und rechte Seite haben folgende Gestalt:

$$\lambda = (\lambda_1, \ldots, \lambda_m, \beta_0, \ldots, \beta_n)^T \quad \text{bzw.} \quad b = (y_2 - \delta, y_2, \ldots, y_m, 0, \ldots, 0).$$

Zieht man die erste Gleichung von der zweiten ab, so erhält man

$$\epsilon^3(\lambda_1 - \lambda_2) + \sum_{i=3}^{m} \lambda_i(\|x_l - x_i\| - \|x_z - x_i\|) + \sum_{i=1}^{n} x_l^i - x_z^i = \delta.$$

Läßt man nun ε gegen null gehen, so geht natürlich x_l gegen x_z. Folglich werden alle Summanden auf der linken Seite immer kleiner, wenn nicht mindestens ein λ^p gegen unendlich geht. Dies muß aber der Fall sein, weil δ auf der rechten Seite nach unten beschränkt ist. Da die Summe über alle λ_i aber null ergibt, muß es auch noch ein weiteres λ^m geben, durch das λ^p zumindest in seiner Größenordnung ausgeglichen wird.
Betrachten wir nun, was unter diesen Voraussetzungen mit

$$\int_{\mathcal{D}} \|H_{\Phi^\varepsilon}(x)\|_F^2 dx$$

passiert. Mit

$$\Phi_\varepsilon(x) = \sum_{i=1}^{m} \lambda_i \|x - x_i\|^3 + \beta_0 + \sum_{i=1}^{n} \beta_i x^i$$

erhalten wir

$$\frac{\partial \Phi_\varepsilon(x)}{\partial x^j} = \sum_{i=1}^{m} \lambda_i \frac{\partial \|x - x_i\|^3}{\partial x^j} + \beta_j = 3 \sum_{i=1}^{m} \lambda_i \|x - x_i\|(x^j - x_i^j) + \beta_j,$$

und daraus

$$\frac{\partial^2 \Phi_\varepsilon(x)}{\partial x^i x^k} = \begin{cases} 3\sum_{i=1}^m \lambda_i \overbrace{\left(\frac{(x^j - x_i^j)^2}{\|x - x_i\|} + \|x - x_i\| \right)}^{h_{j,j}} & \text{wenn } j = k \\ 3\sum_{i=1}^m \lambda_i \underbrace{\left(\frac{(x^j - x_i^j)(x^k - x_i^k)}{\|x - x_i\|} \right)}_{h_{j,k}} & \text{wenn } j \neq k. \end{cases}$$

was zeigt, daß es sich beim Interpolanden um eine zweimal stetig differenzierbare Funktion handelt.
Es ist nicht gelungen, eine Stammfunktion der Frobeniusnorm von $H_{\Phi^\varepsilon}(x)$ zu finden. Trotzdem scheint die Behauptung, daß die Größenordnung des Integrals von der Größenordnung der λ_i abhängt, nicht abwegig. Bis auf x^z sind alle anderen Punkte fix. Nähert man x^z immer mehr an x^l an, so ändern sich nur die $h_{j,k}$, die x^z enthalten. Die $h_{j,k}$, die gleich bleiben, werden mit sich verändernden λ_i multipliziert, wovon mindestens eines gegen unendlich geht. Damit wäre es durchaus plausibel, daß das Integral über alle Grenzen steigt, es erscheint dagegen fragwürdig, ob sich die λ_i mit den Vorfaktoren $h_{j,k}$ über das gesamte Integrationsgebiet immer zufällig aufheben.

4.3 Implementierung und weitere Details

Viele der in den Verfahren verwendeten Ausdrücke müssen erst mühsam berechnet werden. Dies geschah durch eine Implementierung in Matlab. Es sind weiterhin einige Modifikationen üblich, die kurz erläutert sein sollen.
Eine Stammfunktion von (4.5) war nicht zu ermitteln, deswegen wurde diese Größe durch Monte-Carlo-Integration angenähert. Es wurden jeweils 5000 Punkte zur Approximation verwendet, damit konnte man jedes Problem auf einem Rechner mit einem 1MHz Prozessor in ca. drei Tagen lösen.
Die Minimierung des so angenäherten (4.5) erfolgte dadurch, daß $I(x^z)$ zuerst für 2000 zufällig ausgewählte x^z approximiert wurde. Im besten Punkt wurde dann eine Nelder-Mead-Suche gestartet, was möglich ist, da $I(x^z)$ in Matlab implemetiert wurde.
Natürlich kam zur Bestimmung der Φ^z das in Unterabschnitt 1.4.2 beschriebene Vorgehen zum Einsatz. Würde man die Interpolation immer wieder von vorne angehen, wäre das Verfahren nicht durchführbar. Zu Problemen mit schlecht konditionierten Matrizen kam es nicht. Sollten dennoch Probleme auftreten, weil sich

manche Zeilen der Matrix durch sehr eng beieinander liegende Punkte zu sehr ähneln, kann man die Matrix zum Beispiel nach dem in Unterabschnitt 1.4.2 beschriebenen Prinzip vorsichtig aufbauen und immer wieder iterativ nachverbessern, wenn die Interpolationsbedingung in den Knoten zu sehr verletzt wird. Um diese Situation zu vermeiden, gingen jeweils nur die Zentren der verwendeten Differenzensterne in die Interpolation ein, von den übrigen Auswertungen des Sterns wurde nur in der lokalen Suche Gebrauch gemacht.
Alle lokalen Suchen wurden mit der Gitterweite $h = 10^{-4}$ für die Differenzensterne durchgeführt. Dies ergab die besten Werte für das Newton- Verfahren. Die beiden anderen Verfahren lieferten mit anderen Schrittweiten keine besseren Ergebnissen. Nur im Vergleich zum Newton-Verfahren konnte dadurch ein besseres Ergebnis erzielt werden. In dieser Arbeit sollte allerdings mit den bestmöglichen Ergebnissen des Newton-Verfahrens verglichen werden.
Ähnlich wie in den Arbeiten [23] und [33] wurde nicht mit Ξ, Y gerechnet. Weil wir tendenziell wenige Auswertungen haben, wird die Zielfunktion in den Bereichen kleiner Werte sehr schlecht approximiert, wenn zwischen den Zielfunktionswerten mehrere Größenordnung liegen. Auf diese Weise wäre keine vernünftige lokale Suche möglich. In beiden erwähnten Arbeiten wurde deshalb $m_y := \text{median}(Y)$ und alle Zielfunktionswerte, die über m_Y lagen, genau dadurch ersetzt. In dieser Arbeit war die Maßnahme etwas weniger drastisch: Es wurde das 0.75 Quantil q_Y errechnet und alle darüber liegenden Wert durch q_Y ersetzt.

Kapitel 5

Ausblick

Auch wenn das wesentliche Ziel der Arbeit, die Entwicklung eines Verfahrens für den hochparallelen Fall, erreicht wurde, ist die Fragestellung noch nicht erschöpfend bearbeitet. Man könnte noch viel ergänzen.
So ist ein Beweis der globalen Konvergenz für den beliebigdimensionalen Fall wahrscheinlich erreichbar. Zumindest erlauben die Ausführungen in Abschnitt 4.2.3 diese Vermutung, ohne daß man dadurch Unrealistisches erwartet.
Das hochparallele Verfahren könnte noch aufgewertet werden, indem man die lokalen Suchen in den Minima des Interpolanden verbessert. Man könnte dann sicher nicht mehr so viele Minima gleichzeitig untersuchen, wäre dann dort aber wahrscheinlich schneller.
Ein Ziel für die Zukunft ist sicherlich der Test des Verfahrens an realistischen Daten. Dann müßte man sich noch genauer mit den Randbedingungen beschäftigen, auch wenn zu vermuten ist, daß das Verfahren eine ähnliche Leistung bringt wie in den Testbeispielen.
Das Wegfallen einer hier gemachten Vereinfachung der Problemstellung eröffnet eine neue Fragestellung für spätere Arbeiten: Was könnte man tun, wenn nicht alle Variablen hochparallel geändert werden können, sondern nur einige? Welche Auswirkungen hat das auf den Einsatz des Halley-Verfahrens, könnte man es dann überhaupt noch durchführen?
Dieser Ausblick kann, so wie die gesamte Arbeit, nicht den Anspruch der Vollständigkeit erheben. Es gibt auf dem bearbeiteten Feld noch viel zu tun. Deswegen schließen wir mit einem Zitat des Ökonomen und Soziologen Thorstein Veblen.

> The outcome of any serious research can only be to make two questions grow where only one grew before

Anhang

Testprobleme zur lokalen Optimierung

Im folgenden werden 16 Testbeispiele für unrestringierte Optimierungsprobleme angegeben. Sie sind der Testsammlung von von More, Garbow und Hillstrom, siehe [30], entnommen. Alle zu minimierenden Funktionen sind von der Form

$$f(x) = \sum_{i=1}^{m} (F_i(x))^2$$

für $F : \mathbb{R}^n \to \mathbb{R}^m$. Angegeben sind jeweils

(a) die Anzahl n der Variablen x_j und die Anzahl m der Funktionen F_i,

(b) die Funktionen F_i.

1. Rosenbrock-Funktion

a) n gerade, $m = n$

b) $F_{2i-1}(x) = 10(x_{2i} - x_{2i-1}^2)$
$F_{2i}(x) = 1 - x_{2i-1}$

2. Spiralförmiges Tal (Helical valley function)

a) $n = 3$, $m = 3$

b) $F_1(x) = 10[x_3 - 10\theta(x_1, x_2)]$
$F_2(x) = 10[(x_1^2 = x_2^2)^{\frac{1}{2}} - 1]$
$F_3(x) = x_3$

wobei

$$\theta(x_1, x_2) = \begin{cases} \frac{1}{2\pi} \arctan\left(\frac{x_2}{x_1}\right) & \text{falls } x_1 > 0 \\ \frac{1}{2\pi} \arctan\left(\frac{x_2}{x_1}\right) + .5 & \text{falls } x_1 < 0 \end{cases}$$

3. Gauß-Funktion

   (a) $n = 3$, $m = 15$

   (b) $F_i(x) = x_1 \exp[\frac{-x_2(t_i - x_3)^2}{2}] - y_i$
   wobei $t - i = \frac{8-i}{2}$ und

i	y_i
1, 15	0.0009
2, 14	0.0044
3, 13	0.0175
4, 12	0.0540
5, 11	0.1295
6,10	0.2420
7,9	0.3521
8	0.3989

4. Powells schlechtskalierte Funktion

   (a) $n = 2$, $m = 2$

   (b) $F_1(x) = 10^4 x_1 x_2 - 1$
   $F_2(x) = \exp[-x_1] + \exp[-x_2] - 1.0001$

5. Box' dreidimensionale Funktion

   (a) $n = 3$, $m \geq n$ beliebig

   (b) $F_i(x) = \exp[-t_i x_1] - \exp[-t_i x_2] - x_3(\exp[-t_i] - \exp[-10t_i])$
   wobei $t_i = (0.1)^i$

6. Beliebig-dimensionale Funktion

   (a) n beliebig, $m = n + 2$

(b) $F_i(x) = x_i - 1,\ i = 1, \ldots, n$
$F_{n+1}(x) = \sum_{j=1}^{n} j(x_j - 1)$
$F_{n+2}(x) = \left(\sum_{j=1}^{n} j(x_j - 1)\right)^2$

7. Watson-Funktion

(a) $2 \leq n \leq 31,\ m = 31$

(b) $F_i(x) = \sum_{j=2}^{n}(j-1)x_j t_i^{j-2} - \left(\sum_{j=1}^{n} x_j t_i^{j-1}\right)^2 - 1$
wobei $t_i = \frac{i}{29},\ 1 \leq i \leq 29$
$F_{30}(x) = x_1,\ F_{31}(x) = x_2 - x_1^2 - 1$

8. Penalty-Funktion I

(a) n beliebig, $m = 2n$

(b) $F_i(x) = a^{\frac{1}{2}}(x_i - 1),\ 1 \leq i \leq n$
$F_{n+1}(x) = \left(\sum_{j=1}^{n} x_j^2\right) - \frac{1}{4}$
wobei $a = 10^{-5}$

9. Penalty-Funktion II

(a) n beliebig, $m = 2n$

(b) $F_1(x) = x_1 - 0.2$
$F_i(x) = a^{\frac{1}{2}}(\exp[\frac{x_i}{10}] + \exp[\frac{x_{i-1}}{10}] - y_i),\ 2 \leq i \leq n$
$F_i(x) = a^{\frac{1}{2}}(\exp[\frac{x_{i-n+1}}{10}] - \exp[\frac{-1}{10}]),\ n \leq i \leq 2n$
$F_{2n}(x) = \left(\sum_{j=1}^{n}(n-j+1)x_j^2\right) - 1$
wobei $a = 10^{-5}$ und $y_i = \exp[\frac{i}{10}] + \exp[\frac{i-1}{10}]$

10. Browns schlechtskalierte Funktion

(a) $n = 2,\ m = 3$

(b) $F_1(x) = x_1 - 10^6$
$F_2(x) = x_2 - 2\dot{1}0^{-6}$
$F_3(x) = x_1 x_2 - 2$

11. Brown-Dennis-Funktion

(a) $n = 4$, $m \geq n$ beliebig

(b) $F_i(x) = (x_1 + t_i x_2 - \exp[t_i])^2 + (x_3 + x_4 \sin(t_i) - \cos(t_i))^2$
wobei $t_i = \frac{i}{5}$

12. Cox-Funktion

(a) $n = 3$, $n \leq m \leq 100$

(b) $F_i(x) = \exp\left[-\frac{|y_i - x_2|^{x_3}}{x_1}\right] - t_i$
wobei $t_i = \frac{i}{100}$ und $y_i = 25 + (-50 \ln(t_i))^{\frac{2}{3}}$

13. Trigonometrische Funktion

(a) n beliebig, $m = n$

(b) $F_i(x) = n - \sum_{j=1}^{n} \cos x_j + i(1 - \cos x_i) - \sin x_i$

14. Erweiterte singuläre Funktion von Powell

(a) n ein Vielfaches von 4, $m = n$

(b) $F_{4i-3}(x) = x_{4i-3} + 10x_{4i-2}$
$F_{4i-2}(x) = 5^{\frac{1}{2}}(x_{4i-1} - x_{4i})$
$F_{4i-1}(x) = (x_{4i-2} - 2x_{4i-1})^2$
$F_{4i}(x) = 10^{\frac{1}{2}}(x_{4i-3} - x_{4i}^2)$

15. Beale-Funktion

(a) $n = 2$, $m = 3$

(b) $F_i(x) = y_i - x_i(1 - x_2^i)$
wobei $y_1 = 1.5$, $y_2 = 2.25$, $y_3 = 2.625$

16. Wood-Funktion

(a) $n = 4$, $m = 6$

(b) $F_1(x) = 10(x_2 - x_1^2)$
$F_2(x) = 1 - x_1$
$F_3(x) = 90^{\frac{1}{2}}(x_4 - x_3^2)$
$F_4(x) = 1 - x_3$
$F_5(x) = 10^{\frac{1}{2}}(x_2 + x_4 - 2)$
$F_6(x) = 10^{\frac{-1}{2}}(x_2 - x_4)$

Testprobleme zur globalen Optimierung

Im Folgenden werden die Funktionen aufgeführt, wie sie in [14] definiert wurden. Für unsere Rechnungen wurden alle Funktionen linear auf den entsprechenden Würfel $[0,1]^n$ transformiert.

1. Shekel 5, 7, 10 (S5, S7, S10):

$$f(x) = -\sum_{i=1}^{m} \frac{1}{(x-a_i)^T(x-a_i)+c_i},$$

wobei $x = (x_1, \ldots, x_n)^T$, $a_i = (x_{i1}, \ldots, x_{in})^T$, $c_i > 0$
Zulässige Menge: $0 \leq x_j \leq 10$, $j = 1, \ldots n$.
Wir betrachten drei Fälle in der unten stehenden Tabelle mit:
$n = 4$, $m = 5, 7, 10$

i	a_i				c_i
1	4	4	4	4	0.1
2	1	1	1	1	0.2
3	8	8	8	8	0.0
4	6	6	6	6	0.4
5	3	7	3	7	0.4
6	2	9	2	9	0.6
7	5	5	5	3	0.3
8	8	1	8	1	0.7
9	6	2	6	2	0.5
10	7	3.6	7	3.6	0.5

2. Hartmann 3, 6 (H3, H6):

$$f(x) = -\sum_{i=1}^{m} c_i \exp\left(-\sum_{j=1}^{m} a_{ij}(x_j - p_{ij})^2\right)$$

wobei $x = (x_1, \ldots, x_n)^T$, $p_i = (p_{i1}, \ldots, p_{in})^T$, $a_i = (a_{i1}, \ldots, a_{in})^T$.
Zulässige Menge: $0 \leq x_i \leq 1$

a) $m = 4$, $n = 3$

i	a_{ij}			c_i	p_{ij}		
1	3	10	30	1	0.3689	0.1170	0.2673
2	0.1	10	35	1.2	0.4699	0.4387	0.7470
3	3	10	30	3	0.1091	0.8732	0.5547
4	0.1	10	35	3.2	0.03815	0.5743	0.8828

b) $m = 4$, $n = 6$

i	a_{ij}						c_i
1	10	3	17	3.5	1.7	8	1
2	0.05	10	17	0.1	8	14	1.2
3	3	3.5	1.7	10	17	8	3
4	17	8	0.05	10	0.1	14	3.2

i	p_{ij}					
1	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650
4	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

3. Branin

$$f(x_1, x_2) = a(x_2 - bx_1^2 + cx_i - d)^2 + e(1 - f)\cos x_1 + e$$

mit $a = 1$, $b = 5.1(4\pi^2)$, $c = \frac{5}{\pi}$, $d = 6$, $e = 10$, $f = \frac{1}{8\pi}$.
Zulässiger Bereich: $-5 \leq x_1 \leq 10$, $0 \leq x_2 \leq 15$.

4. Goldstein & Price

$$\begin{aligned} a &= [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \\ b &= [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \\ f(x_1, x_2) &= a * b \end{aligned}$$

Zulässiger Bereich: $-2 \leq x_{1,2} \leq 2$.
Es existieren vier lokale Minima. Das globale Minimum befindet sich in $(0, -1)$ mit Funktionswert 3.

Literaturverzeichnis

[1] P. Alfeld, *A trivariate Clough-Tocher scheme for tetrahedral data.*, Comput. Aided Geom. Des., 1 (1984), pp. 169–181.

[2] C. Barber, *The quickhull algorithm for convex hulls*, ACM Trans. Mathematical Software, 22 (1996), pp. 469 – 483.

[3] M. Bjoerkman and K. Holmstroem, *Global optimization of costly nonconvex functions using radial basis functions.*, Optim. Eng., 1 (2000), pp. 373–397.

[4] M. D. Buhmann, *Radial Basis Functions: Theory and Implementations*, Cambridge University Press, 2003.

[5] A. S. Cavaretta, C. A. Micchelli, and A. Sharma, *Multivariate interpolation and the Radon transform.*, Math. Z., 174 (1980), pp. 263–279.

[6] E. K. P. Chong and S. H. Zak, *An introduction to optimization. 2nd ed.*, Chichester: Wiley. xv, 2001.

[7] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North Holland, 1979.

[8] I. Crain, B.K. Bhattacharyya, *Treatment of non-equispaced two-dimensional data with a digital computer*, Geoexploration, 5 (1967), pp. 173 – 194.

[9] G. Cressmann, *An operational objective analysis system*, Monthly Weather Review, 87 (1959), pp. 367 – 374.

[10] A. A. Cuyt and L. Rall, *Computational implementation of the multivariate Halley method for solving nonlinear systems of equations.*, ACM Trans. Math. Softw., 11 (1985), pp. 20–36.

[11] C. de Boor, *A Practical Guide to Splines*, Springer, 1978.

[12] C. de Boor and A. Ron, *On multivariate polynomial interpolation.*, Constructive Approximation, 6 (1990), pp. 287–302.

[13] C. De Boor and A. Ron, *Computational aspects of polynomial interpolation in several variables.*, Math. Comput., 58 (1992), pp. 705–727.

[14] L. Dixon and G. Szego, *Towards global optimisation 2.*, Amsterdam - New York - Oxford: North-Holland Publishing Company. , 1978.

[15] J. Duchon, *Splines minimizing rotation-invariant semi-norms in sobolev spaces*, in Constructive Theory of Functions of Several Variables, Springer, 1977, p. 174.

[16] B. Fornberg, *Generation of finite difference formulas on arbitrary spaced grids.*, Math. Comput., 51 (1988), pp. 699–706.

[17] M. Gasca, T. Sauer, *On the history of multivariate polynomial interpolation*, Journal of Computational and Applied Mathematics, 122 (2000), pp. 23 – 35.

[18] ——, *Polynomial interpolation in several variables*, Advances in Computational Mathematics, 12 (2000), pp. 377 – 410.

[19] C. Geiger and C. Kanzow, *Numerical methods for solution of unconstrained optimization problems. (Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben.)*, Berlin: Springer. xii, 350 S, 1999.

[20] P. George, H. Borouchaki, *Delaunay Triangulation and Meshing*, Hermes, 1998.

[21] T. Goodman, *Interpolation in minimum semi-norm and multivariate B-splines.*, J. Approximation Theory, 37 (1983), pp. 212–223.

[22] W. Gordon, J.A. Wixom, *Shepard's method of „metric interpolation"to bivariate und multivariate interpolation*, Mathematics of Computation, 32 (1978), pp. 253 – 264.

[23] H. Gutmann, *A radial basis function method for global optimization*, Journal of Global Optimization, 19(3) (2001), pp. 201 – 227.

[24] J. Hoschek, D. Lasser, *Grundlagen der geometrischen Datenverarbeitung*, Teubner, 1992.

[25] D. Jones, C. Perttunen, and B. Stuckman, *Lipschitzian optimization without the Lipschitz constant.*, J. Optimization Theory Appl., 79 (1993), pp. 157–181.

[26] D. R. Jones, *Efficient global optimization of expensive black-box-functions*, Journal of Global Optimization, 13 (1998), pp. 455 – 492.

[27] ——, *A taxonomy of global optimization methods based on response surfaces*, Journal of Global Optimization, 21 (2001), pp. 345 – 383.

[28] P. Kergin, *A natural interpolation of c^k functions*, Journal of Approximation Theory, 29 (1980), pp. 278 – 293.

[29] M. Locatelli, *Bayesian algorithms for one-dimensional global optimization.*, J. Glob. Optim., 10 (1997), pp. 57–76.

[30] J. J. More, B. S. Garbow, and K. E. Hillstrom, *Testing unconstrained optimization software.*, ACM Trans. Math. Softw., 7 (1981), pp. 17–41.

[31] M. Powell, *A review of algorithms for thin plate spline approximation*, in Advanced Topics in Multivariate Approximation, World Scientific, 1996, pp. 303–322.

[32] M. Powell, M. Sabin, *Piecewise quadratic approximations on triangles*, ACM Trans. Math. Software, 3 (1977), pp. 316 – 325.

[33] R. G. Regis and C. A. Shoemaker, *Constrained global optimization of expensive black box functions using radial basis functions.*, J. Glob. Optim., 31 (2005), pp. 153–171.

[34] P. Sablonniere, *Composite finite elements of class c^k.*, Journal of Computational and Applied Mathematics, 12 13 (1985), pp. 541 – 550.

[35] J. Sacks, *Design and analysis of computer experiments (with discussion)*, Statistical Science, 4 (1989), pp. 405 – 435.

[36] P. Spellucci, *Numerische Verfahren der nichtlinearen Optimierung*, Birkhäuser Verlag, Basel, Boston, Berlin, 1993.

[37] J. Stoer, *Numerical mathematics. 1: An introduction - under consideration of lectures by F. L. Bauer. (Numerische Mathematik. 1: Eine Einführung - unter Berücksichtigung von Vorlesungen von F. L. Bauer.) 8., neu bearb. u. erw. Aufl.*, Berlin: Springer, 1999.

[38] A. Torn and A. Zilinskas, *Global optimization.*, Lecture Notes in Computer Science. 350. Berlin etc.: Springer-Verlag. X, 255 p. , 1989.

[39] A. Worsey and G. Farin, *An n-dimensional clough-tocher interpolant*, Constructive Approximation, 3 (1987), pp. 99–110.

[40] A. Worsey and B. Piper, *A trivariate powell-sabin interpolant*, Comput. Aided Geom. Des., 5 (1988), pp. 177–186.

[41] A. Zenisek, *Polynomial approximation on tetrahedrons in the finite element method.*, J. Approximation Theory, 7 (1973), pp. 334–351.

[42] ——, *A general theorem on triangular finite C^m-elements*, Revue Francaise d'Automatique, d'Informatique et de Recherche Oprationelle, 2 (1974), pp. 119 – 127.

Akademischer Werdegang

Michael Godzierz

geboren am 28. Juli 1977 in Bensheim

1984 - 1988	Joseph-Heckler-Grundschule, Bensheim
1988 - 1997	Altes-Kurfürstliches-Gymnasium, Bensheim
1997	Abitur
1997 - 1998	Zivildienst
1998 - 2003	Studium Diplom Mathematik mit Nebenfächern Informatik und Physik an der Technischen Universität Darmstadt
2001 - 2002	Studienaufenthalt an der Tulane University New Orleans in den USA
Oktober 2000	Vordiplom
Ausgust 2002	Master of Science
Mai 2003	Diplom
Seit Mai 2003	Wissenschaftlicher Mitarbeiter und Doktorand am Fachbereich Mathematik der Technischen Universität Darmstadt